NONLINEAR PREDICTIVE CONTROL USING GENETIC ALGORITHMS

BY

WASIF NAEEM

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

MAY 2001

UMI Number: 1407215

UMI®

UMI Microform 1407215

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by

WASIF NAEEM

under the direction of his Thesis Advisor and approved by his Thesis Committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

Thesis Committee

Dr. AL – DUWAISH HUSSAIN N. (Chairman)

Dr. AL – BAIYAT SAMIR A. (Member)

Dr. ABDEL – MAGID Y.L. (Member)

Department Chairman

Dean of Graduate Studies

22-07-2001

Date

Dedicated

to

my beloved family and especially
in the memory of my late Baijee

Acknowledgements

In the beginning, I must thank to Allah, the Almighty and the Most Merciful for His blessings throughout my life in general and in the course of this thesis in particular.

I am thankful to King Fahd University of Petroleum and Minerals for supporting my M.S. studies and this research work.

I would like to pay a high tribute to my thesis advisor Dr. Hussain N. Al-Duwaish for his invaluable guidance and helpful ideas throughout this thesis work. His appreciation and words of encouragement gave a new life to my efforts in hard times. I am also indebted to him for his valuable time, efforts and his continuous support and inspiration. At the later stages, he helped me a lot in writing my thesis.

I would also like to express my deep appreciation to my committee members, Dr. Samir Al-Baiyat and Dr. AbdelMagid, for their constant help and encouragement. I am thankful to Dr. Sameer Al-Baiyat, Chairman Electrical Engineering Department for establishing an excellent graduate research laboratory in our department.

I am thankful to Mr. Ahmet Palazoglu from the Department of Chemical Engineering and Materials Science at the University of California, Davis for providing me some very useful papers.

I am also thankful to my old friends Tanveer, Imran, Zeeshan and Moiz who are not here with me, and also to my friends which i made at KFUPM especially Moin, Saad, Ahmer, Atif, Asif, Fareed, Ajmal, Junaid, Sajid, Shah and Shafayat for their moral support and guidance throughout my stay at KFUPM. Thanks are also due to Raslan for helping me in the Arabic abstract of this thesis.

In the last, but not the least, I am thankful to my parents, my brother, my sister and my brother-in-law for their love, guidance and support.

Contents

List of Tables

List of Figures

THESIS ABSTRACT

Name: WASIF NAEEM

Title: NONLINEAR PREDICTIVE CONTROL USING GENETIC ALGORITHMS

Degree: MASTER OF SCIENCE

Major Field: ELECTRICAL ENGINEERING

Date of Degree: MAY 2001

In recent years, the requirements for the quality of automatic control in the process industries increased significantly due to increased complexity of the plants and sharper specifications of product. At the same time, the available computing power increased to a very high level. As a result, computer models that were computationally expensive became applicable even to rather complex problems. Model Predictive Control techniques were developed to obtain tighter control and it was introduced successfuly in several industrial plants. MPC can provide robust control for processes with variable gain and dynamics, multivariabe interaction, measured loads, and unmeasured disturbances. The implementation of MPC requires the solution of a constrained optimization problem at each sampling time. Different optimization techniques including linear programming (LP), quadratic programming (QP) and dynamic programming (DP) have been used in MPC.
In this thesis a new approach for the implementation of MPC will be proposed using Genetic Algorithm GA. The proposed method formulates the MPC as an optimization problem and genetic algorithms are used in the optimization process. The advantages of using genetic algorithms include: applicability to any process model, possibility of defining any control objective and capability of handling any process constraints. The proposed method is applied to both SISO and MIMO systems with different types of process models, disturbance models, cost functions and process constraints. Application of the proposed algorithm to chemical processes is emphasized.

Keywords: Model Predictive Control, Genetic Algorithms, Nonlinear Chemical Processes, Optimization, Adaptive Model Predictive Control, Prediction Horizon, Control Horizon.

Master of Science Degree

King Fahd University of Petroleum and Minerals, Dhahran.

MAY 2001

ملخص الرّسالة

ا لا سم: واصف نعيم

عنوا ن الدراسة: التحكم بالتنبؤ الغير خطي باستعمال الخوارزم الجينيّة

التخصص: هندسة الحا سب اللآلي

تاريخ التخر ج: مايو 2001

في السنوات الأخيرة،إزداد الإحتياج لجودة التحكّم الأوتوماتيكيّ في الصناعة العملية بشكل كبير، نتيجة لإزدياد تعقيد المصانع و دقة المواصفات. في الوقت نفسه، إزدادت المقدرة الحسابية للمعالجات لمستوى مرتفع جداً، مما يعني أن نماذج الكمبيوتر التي كانت غالية حسابياً، أصبح بالإمكان إستعمالها حتى في المشاكل المعقدة. تم تطوير تقنيات التكحم لتنبّؤ النماذج للحصول على تحكّم أكثر إحكامًا (MPC: Model Predictive Control). وتم إستعمال هذه التقنيات فعلا في الكثير من المصانع. يوفر التحكم القوي مع الزيادة المتغيرة والديناميكية،والتّفاعل المتعدّد المتغيّرات،و الأحمال الموزونة، والإزعاج غير قابل للقياس. تطبيق ال(MPC) يحتاج لحل مشكلة التّحسين المرغمة في كل عينة من الوقت. يوجد تقنيات للتحسين مختلفة كالبرمجة الخطية، و البرمجة التربيعية، والبرمجة الديناميكية تم استخدامها في ال(MPC). في هذه الرسالة نستخدم تقنية جديدة لحل المشكلة وهي الخوارزم الجينية. الطريقة المقترحة تصيغ ال (MPC) كمشكلة تحسينية والخوارزم الجينية تستعمل في عملية التحسين. الميزات لإستعمال الخوارزم الجيني تتضمن: إمكانية التطبيق في أي نموذج عملي، إمكانية تعريف أي هدف للتحكم، إمكانية معالجة أي قيود. الطريقة المقترحة تم تطبيقها في النظامين (SISO) و (MIMO) مع أنواع مختلفة من النماذج العملية، النماذج الإزعاجية، وظائف التّكلفة و القيود. وتم مناقشة التطبيق للطريقة المقترحة على العمليات الكيميائية.

درجة الماجستير في العلوم

جامعة الملك فهد للبترول و المعادن

الظهران ، المملكة العربيّة السّعوديّة

مايو 2001

Chapter 1

Introduction

1.1 Motivation or Why need MPC?

The PID controller has been a workhorse in process industries for over forty years. PID-type controllers are routinely used in single-input single-output applications with good results, but success with this type of controller for multivariable systems has been limited.

It is now recognized that the limitations of the PID-type controller can be traced to its characteristics. The PID controller came into existence on the basis of hardware realizability [1]. However, with the evolution of digital computers, much better designs can be produced without any consideration for hardware realizability. This, in part, has spurred research and development to evolve better control strategies for process systems.

The second source of incentives for the development of better control systems design procedures lies in the demands of times. Processes today are much more complex. requiring high level of steady state optimization and good closed-loop control. A major source of complexity in process plants is the existence of interacting multivariable systems. Good control of these systems is important since they have a strong influence on smooth operability and overall plant economics. Rising costs of energy and raw materials combined with the availability of powerful. low-cost microprocessers for control have created an additional incentive to evolve better techniques for multivariable process control.

Furthermore it is generally accepted that the most effective way to generate the most profit out of our plants while responding to marketplace variations with minimal capital investment is provided by the integration of all aspects of automation of the decision making process.

Although maintaining a stable operation of the process was possibly the only objective of control systems in the past, this integration imposes more demanding requirements. In addition, the implementation of such integrated systems is forcing our processes to operate over an even wider range of conditions. As a result, the control problem that any control system must solve can be stated as follows [2].

Online update the manipulated variables to satisfy multiple, changing performance criteria in the face of changing plant characteristics.

The whole spectrum of process control methodologies in use today is faced with the solution of this problem. The difference between these methodologies lies in the particular assumptions and compromises made in the mathematical formulation of performance criteria and in the selection of a process representation. These are made primarily to simplify the mathematical problem so that its solution fits the existing hardware capabilities. The natural mathematical representation of many of these criteria is in the form of dynamic objective functions to be minimized and of dynamic inequality constraints. The usual mathematical representation for the process is a dynamic model with its associated uncertainties. The importance of uncertainties is increasingly being recognized by control theoreticians and thus are being included explicitly in the formulation of controllers. However, one of the most crucial compromises made in control is to ignore constraints in the formulation of the problem.

It is a fact that in practice the operating point of a plant that satisfies the overall economic goals of the process will lie at the intersection of constraints. Therefore, in order to be successful, any control system must anticipate constraint violations and correct for them in a systematic way: violations must not be allowed while keeping the operation as close as possible to these constraints. The usual practice in process control is to ignore the constraint issue at the design stage and then handle it in an ad hoc way during the implementation.

Model Predictive Control (MPC) techniques provide the only methodology to handle constraints in a systematic way during the design and implementation of the controller. Moreover, in its general form MPC is not restricted in terms of the model, objective function and constraint functionality. For these reasons, it is the only methodology that currently can reflect most directly the many performance criteria of relevance to the process industries and is capable of utilizing any available process model [2]. This is the primary reason for the success of these techniques in numerous applications in the chemical process industries.

1.2 Historical Background

The current interest of the processing industry in MPC can be traced back to a set of papers which appeared in the late 1970s. In 1978 Richalet. Rault. Testud and Papon described successful applications of "Model Predictive Heuristic Control" [3], and in 1979 engineers from Shell (Cutler and Ramaker). outlined "Dynamic Matrix Control" (DMC) [4], and reported applications to a fluid catalytic cracker. Then in 1987 Clarke *et al.* presents the "Generalized Predictive Control" [5], [6]. In all these algorithms an explicit dynamic model of the plant is used to predict the effect of future actions of the manipulated variables on the output (thus the name "Model Predictive Control"). The future moves of the manipulated variables are determined by optimization with the objective of minimizing the predicted error subject to operating constraints. The optimization is repeated at each sampling time based on updated information (measurements) from the plant.

The success of MPC technology as a process control paradigm can be attributed to three important factors. First and foremost is the incorporation of an explicit process model into the control calculation. This allows the controller, in principle, to deal directly with all significant features of the process dynamics. Secondly the MPC algorithm considers plant behavior over a future horizon in time. This means that the effects of feedforward and feedback disturbances can be anticipated and removed, allowing the controller to drive the plant more closely along a desired future trajectory. Finally the MPC controller considers process input, state and output constraints directly in the control calculations. This means that constraint violations are far less likely, resulting in tighter control at the optimal constrained steady-state for the process. It is the inclusion of constraints that most clearly distinguishes MPC from other process control paradigms [7]. Some good reviews on model predictive control can be found in for example [8], [9], [10], [11], [12], [13] and [14].

Besides DMC and MPHC, there are several other commercially available model predictive controllers available today, some of them are

- MAC (Model Algorithmic Controller) or MPHC (Model Predictive Heuristic Controller)
- GPC (Generalized Predictive Controller)
- EPSAC (Extended Prediction Self Adaptive Controller)
- PFC (Predictive Functional Controller)
- EHAC (Extended Horizon Adaptive Control)

to name a few.

1.3 Linear vs Nonlinear

In this thesis, Model Predictive control is applied to nonlinear processes. Though manufacturing processes are inherently nonlinear, the vast majority of MPC applications to date are based on linear dynamic models, the most common being step and impulse response models derived from the convolution integral. There are several potential reasons for this. Linear empirical models can be identified in a straightforward manner from process test data. In addition, most applications to date have been in refinery processing, where the goal is largely to maintain the process at a desired steady state (Regulator Problem), rather than moving rapidly from one set point to another (Servo Problem). A careful identified linear model is sufficiently accurate in the neighborhood of a single operating point for such applications, especially if high quality feedback measurements are available. For these reasons, in many cases, a linear model will provide the majority of the benefits possible with MPC technology [7].

Nevertheless, there are cases where nonlinear effects are significant enough to justify the use of Nonlinear MPC technology. These include atleast two broad categories of applications:

- Regulator control problem where the process is highly nonlinear and subject to large frequent disturbances.
- Servo control problems where the operating points change frequently and span a sufficiently wide range of nonlinear process dynamics.

It is because of this motivation, nonlinear processes are chosen for this thesis. The next section briefly explains the basic model predictive control algorithm.

1.4 Model Predictive Control Algorithm

In a model predictive control, the process output is predicted by using a model of the process to be controlled. Any model that describes the relationship between the input and the output of the process can be used. Further if the process is subject to disturbances, a disturbance or noise model can be added to the process model, thus allowing the effect of the disturbances on the predicted process output to be taken into account. In order to define how well the predicted process output tracks the reference trajectory, a criterion function is used. Typically the criterion is the difference between the predicted process output and the desired reference trajectory. A simple criterion function [15] is

$$J = \sum_{i=1}^{H_p} [\hat{y}(k+i) - w(k+i)]^2 \tag{1.1}$$

where $\hat{y}$ is the predicted process output, w is the reference trajectory, and H_p is the prediction horizon or output horizon. The structure of an **MPC** is shown in Figure 1.1.

Now the controller output sequence u_{opt} over the prediction horizon is obtained by minimization of J with respect to u. As a result the future tracking error is minimized. If there is no model mismatch i.e., the model is identical to the process and there are no disturbances and constraints, the process will track the reference trajectory exactly on the sampling instants.

For this thesis, Genetic Algorithms (GAs) are chosen for the optimization of the

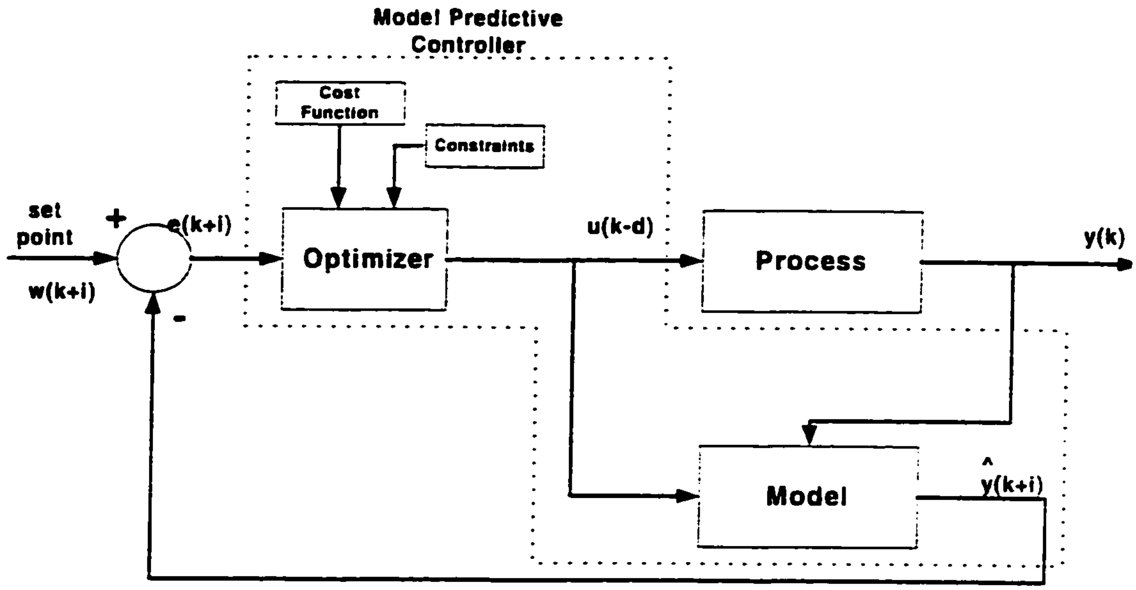

Figure 1.1: Structure of MPC.

performance index in nonlinear MPC. There has been not much work done on non-linear MPC because nonlinear processes are usually linearized about the set point. Some recent papers and surveys on nonlinear MPC are [16], [17], [18], [19], [20], [21] and [7]. The next section describes Genetic Algorithms in detail and the advantages of GAs over other optimization techniques.

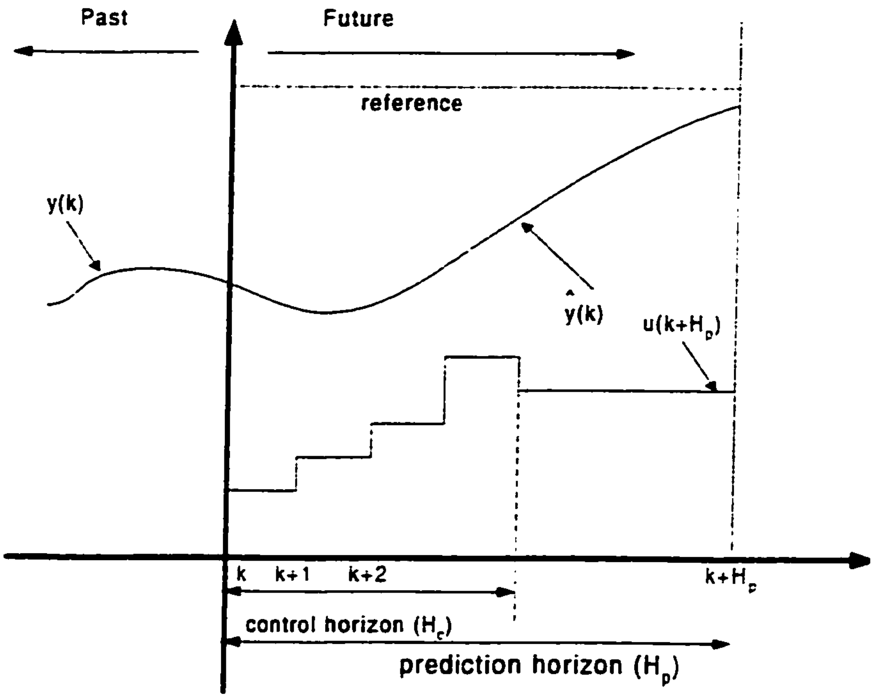

Figure 1.2: Predicted output and the corresponding optimum input over a horizon H_p, where $u(k)$ is the optimum input, $\hat{y}(k)$ is the predicted output, and $y(k)$ is the process output.

1.5 Genetic Algorithms and other Optimization Techniques

The computation of control moves in MPC involve the solution of a constrained optimization problem where analytical solution is usually not available. These computations are obtained using numerical solution of the constrained optimization problem at each sampling time, which necessitates the use of efficient optimization techniques. Quadratic programming (QP) has been used to solve the MPC optimization problem [22]. Dynamic Programming is also used to solve the MPC optimization problem. but all these approaches deals with a protoype objective function. MPC applications using Neural Networks has also been reported in [23] and [24] which requires complex training process. In contrast, Genetic Algorithms has the advantage of using any type of objective function and also has the capability to deal with any type of process model and process constraints. It also avoids the complex training process of the neural networks and fuzzy algorithms [25]. Genetic Algorithms are proven to be be successful in many control system designs. see for example [26]. [25]. [27] and [28] to name a few. The next subsection describes a simple genetic algorithm.

Genetic Algorithms

Genetic Algorithms (GAs) inspired by Darwinian theory. is a powerful non-deterministic iterative search heuristic. Genetic Algorithms operate on a population consists of encoded strings, each string represents a solution. Crossover operator is used on these strings to obtain the new solutions which inherits the good and bad properties of their parent solutions. Each solution has a fitness value. solutions having higher fitness values are most likely to survive for the next generation. Mutation operator is applied to produce new characteristics, which are not present in the parent solutions. The whole procedure is repeated until no further improvement is observed or run time exceeds to some threshold [29].

The flowchart of a simple genetic algorithm is presented in Figure 1.3 and the operation of the *GA* is explained as follows.

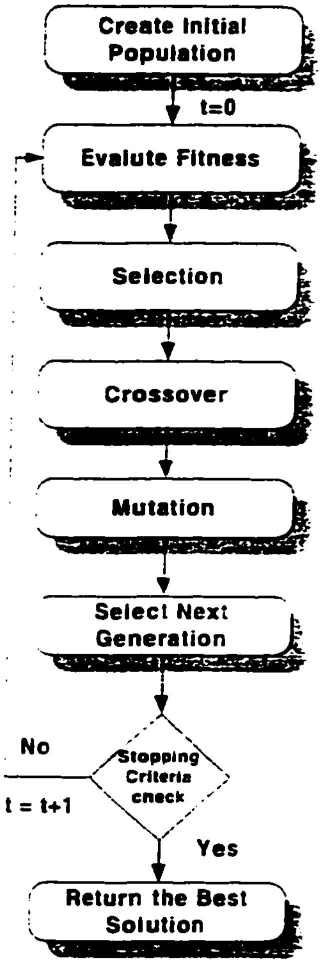

Figure 1.3: Flowchart of a simple Genetic Algorithm.

To start the optimization. GA use randomly produced initial solutions. This method is preferred when a priori knowledge about the problem is not available. After randomly generating the initial population of say N solutions, the GA use the three genetic operators to yield N new solutions at each iteration. In the selection operation, each solution of the current population is evaluated by it's fitness normally represented by the value of some objective function.

Based on the fitness value, individuals are selected for crossover. Different selection methods such as *roulette wheel selection* and *stochastic universal sampling* can be used.

The crossover operator works on pairs of selected solutions with certain crossover rate. The crossover rate is defined as the probability of applying crossover to a pair of selected solutions. There are many ways of defining this operator such as single point crossover . double point crossover, multi-point crossover etc.

Mutation is a random alteration with small probability of the binary value of a string position. This operator prevents *GA* from being trapped in a local minima.

The fitness evaluation unit in *GA* acts as an interface between the *GA* and the optimization problem. Information generated by this unit about the quality of different solutions is used by the selection operation in the *GA*. Next the stopping criteria must be decided. This may be the case when there is no significant improvement in maximum fitness or the maximum allowable time (number of iterations) is passed. At the end of the algorithm, we return the result as the best solution found so far. Now we will explore the basic GA operators in detail.

1.6 Statement of the Problem

This thesis deals with Model Predictive Control of nonlinear processes using Genetic Algorithms. Model predictive control as described before requires the optimization of some quadratic objective function. The optimization is performed by using the GAs. The advantage of using GAs includes: applicability to any process model, possibility of defining any control objective, and capability of handling any process constraint. Moreover, as opposed to other optimization techniques, it is capable of providing global minima.

Some penalty terms are added to the objective objective in addition to the simple objective function given by Equation 1.1. These penalty terms includes the weighted sum of the inputs over the prediction horizon and the weighted sum of the rate of change of inputs over the control horizon. Constraints are taken into account which are due to physical limitations of the actuators. The modified objective function is given by

$$\begin{aligned} J = \sum_{i=1}^{H_p} e(k+i)^T Q e(k+i) \quad &+ \quad \sum_{i=1}^{H_c} \Delta u(k+i)^T R \Delta u(k+i) \\ &+ \quad \sum_{i=1}^{H_p} u(k+i)^T S u(k+i) \end{aligned} \tag{1.2}$$

subject to

$$u^l \leq u(k+i) \leq u^u$$

$$\Delta u^l \leq \Delta u(k+i) \leq \Delta u^u$$

$$y^l \leq y(k+i) \leq y^u$$

where the superscripts l and u represents the lower and upper bounds respectively. Q is the weight on the prediction error

$$e(k) = \hat{y}(k) - w(k)$$

where w(k) is the reference or the desired set point. R and S are weights on the change in the input Δu, and magnitude of the input u respectively.

Different types of process models are used in this thesis. The focus is mainly on *Hammerstein* and *Wiener models*. Although some linear examples are taken but they are just motivating examples. The process models used in this thesis are both single-input single-output (SISO) and multi-input multi-output (MIMO). A nonlinear model involving the noise term is also included.

Model uncertainty is the major cause of producing instability in Model Predictive Controllers. It is highly unlikely that the model is exactly equivalent to the plant. therefore real time implementation of the proposed algorithm is investigated and a solution is proposed for it by using simultaneous online control and identification.

1.7 Thesis Organization

This thesis is organized as follows. In the first chapter. motivation is provided as to why MPC is so important. A brief description of the algorithm is presented. Genetic algorithms are explored and advantages of genetic algorithms over other optimization techniques are provided.

In the second chapter. MPC is explored in detail. Specifically the two pioneer MPC technologies. IDCOM (now MAC). and DMC are presented. Difference between different predictive controllers are also provided.

In the third chapter. different process models used in model predictive controllers are described. Linear and nonlinear models are described separately with main emphasis on nonlinear models.

In the fourth chapter. simulation results are presented. The proposed algorithm is applied to different process models. Both regulator and servo problems are considered. Real time implementation of the proposed algorithm is also investigated.

In the last chapter. summary of the thesis and future work is presented.

Chapter 2

Model Predictive Control

2.1 Introduction

In Chapter 1, a brief introduction of Model Predictive Control was provided. A superficial overview of how the algorithm works was also presented. In this chapter, MPC is explored in detail. A comprehensive discussion of the algorithm is provided, along with the strategies used by some well known controllers like $MPHC$ and DMC. Some advantages and disadvantages are also dicussed.

The next section provides some basic concepts related to MPC technology.

2.2 The Predictive Controller Concept

Model Predictive Control (MPC) refers to a class of algorithms that compute a sequence of manipulated variable adjustments in order to optimize the future behavior of a plant. The generation of the optimal sequence is done by using a model of the process. As said before, any model that describes the input and output relation of a process can be used. Hence, not only transfer-function models can be used, but also step-response models, state-space models and nonlinear models. Further, if the process is subject to disturbances, a disturbance or noise model can be added to the process model, thus allowing the effect of disturbances on the predicted process output to be taken into account [15]. A detailed description of different types of linear and nonlinear models can be found in Chapter 3.

Now consider again Figure 1.2. Suppose that the current time is denoted by k. Further $u(k), y(k)$ and $\hat{y}(k)$ denote the controller output, the process output and the predicted process output at time k, respectively. Now, define

$$u = [u(k), u(k+1), \cdots, u(k+H_p-1)]^T \quad (2.1)$$

$$\hat{y} = [\hat{y}(k+1), \hat{y}(k+2), \cdots, \hat{y}(k+H_p)]^T \quad (2.2)$$

$$w = [w(k+1), w(k+2), \cdots, w(k+H_p)]^T \quad (2.3)$$

where w is the desired process output or set point and H_p is the prediction horizon i.e., the time upto which the output is predicted in the future. Then, a predictive controller calculates such a future controller output sequence u shown in Figure 1.2, such that the predicted output of the process $\hat{y}$ is close to the desired process output w. This desired process output is called the *reference trajectory*.

Rather than using the complete controller output sequence determined in the above way to control the process in the next H_p samples, only the first element of this controller output sequence $u(k)$ is used to control the process. At the next sample. $k+1$, the whole procedure is repeated using the latest measured information. This is called *receding horizon* principle. Assuming that there are no disturbances and no modeling error, the predicted process output $\hat{y}(k+1)$ is exactly equal to the process output. Now, again, a future controller output sequence is calculated such that the predicted process output is close to the reference trajectory. In general, this controller output sequence is different from the one obtained at the previous sample. The reason for using the receding horizon approach is that it allows us to compensate for future disturbance or modelling errors. For example, due to a disturbance or modelling error the predicted process output $\hat{y}(k+1)$ is not equal to the process output $y(k)$. Then, it is intuitively clear that at time $k+1$, it is better to start the predictions from the measured process output rather than from the process output predicted at the previous sample. The predicted process output is now corrected for disturbances and modelling errors. A feed-back mechanism is activated. As a result of the receding horizon approach the horizon over which the process output is predicted shifts one sample into the future at every sample instant. In order to define how well the predicted process output tracks the reference trajectory, a criterion function is used. Typically, such a criterion function is a function of $\hat{y}, w$ and u. A simple criterion function is

$$J = \sum_{i=1}^{H_p} [\hat{y}(k+i) - w(k+i)]^2 \qquad (2.4)$$

In some controllers, the criterion function is augmented with different penalty terms usually involving the input u and the rate of change of the input Δu, which are used to penalize them, when they exceed some desired threshold. The criterion function is thus

$$J = \sum_{i=1}^{H_p} e(k+i)^T Q e(k+i) + \sum_{i=1}^{H_c} \Delta u(k+i)^T R \Delta u(k+i) + \sum_{i=1}^{H_p} u(k+i)^T S u(k+i) \tag{2.5}$$

where Q, R and S are the weighting matrices, and ϵ represents the error between the desired output and the predicted output.

$$\epsilon = w(k) - \hat{y}(k)$$

Now the controller output sequence u_{opt} over the prediction horizon is obtained by minimization of J with respect to u :

$$u_{opt} = \min_{u} J$$

Then, u_{opt} is optimal with respect to the criterion function that is minimized. As a result, the future tracking error is minimized. If the model is identical to the process and there are no disturbances and constraints, the process will track the reference trajectory exactly on the sampling instants. In summary, Model Predictive Control algorithm, consists of the following three steps [10].

1. Explicit use of a model to predict the process output along a future time horizon (Prediction Horizon).

2. Calculation of a control sequence along a future time horizon (Control Horizon), to optimize a performance index.

3. A receding horizon strategy, so that at each instant the horizon is moved towards the future, which involves the application of the first control signal of the sequence calculated at each step which is illustrated in Figure 1.2.

2.3 Process Constraints

For a constrained model predictive control of a physical system, some criteria must be satisfied along with the minimization of the quadratic objective function. These conditions are referred to as *Constraints*. The process becomes nonlinear because of the presence of these constraints, even if it is linear. The most common constraints are on the input of the process, or equivalently on the output of the controller. These constraints are due to the physical limitation of the actuators. Constraints can be categorized into two types commonly used in MPC technology [7].

1. Hard Constraints

2. Soft Constraints

In hard constraints, no violation of the bounds are allowed at any time. Hard constraints are usually imposed on the input to the process. For soft constraints, violation of the bounds can be allowed temporarily for the satisfaction of other criteria, the magnitude of the violation is generally subjected to a quadratic penalty in the objective function. Soft constraints are usually implemented on the output of the process. The use of hard output constraints is generally avoided in MPC because a disturbance can easily cause such a controller to lose feasibility [7]. Various types of hard and soft constraints are:

- Equality Constraints
- Non-Equality Constraints
- End Constraints
- Level Constraints
- Rate Constraints

Although more or less these constraints are the same but these are the terminologies which are most frequently used in the literature so it is of worth mentioning them.

Equality Constraints

Equality constraints refers to the equality of some input or output to a specified value. e.g., [30]. This constraint can be implemented in regulator problems. where the process output needs to be maintained at a fix value.

$$k(x(t), u(t), t) = 0$$

Inequality Constraints

It refers to the condition that the input or output must be greater or less than some specified value, e.g., [30].

$$h(x(t), u(t), t) \geq 0$$

End Constraints

The constraints imposed at the end of the (finite) prediction horizon are referred to as End Constraints.

Level Constraints

This constraint is the aggregation of equality and inequality constraints. It refers to the condition when the controller is restricted between two values: the upper limit u_{max} and lower limit u_{min}, i.e.,

$$u_{min}(k) \leq u(k) \leq u_{max}(k) \tag{2.6}$$

Rate Contraints

It refers to the condition when the change of the controller output per sample is limited between two values. This is usually done to avoid large changes in input moves to limit large changes in the output of the process.

$$\Delta u_{min}(k) \leq \Delta u(k) \leq \Delta u_{max}(k) \tag{2.7}$$

Some papers dealing with constrained MPC are for example [22], [31] and [32].

2.4 DMC, MAC, GPC and other Predictive Controllers

Although the basic idea underlying all predictive controllers is the same, they all differ in their details. Some of the predictive controllers found in industry are mentioned in Chapter 1. In this section, MAC is explored in detail, while some basic properties are provided for other controllers like GPC and DMC.

2.4.1 Model Predictive Heuristic Control(MPHC)

Model Predictive Heuristic Control invented by J. Richalet and his companions in 1978 boosts the research in the field of predictive controller. MPHC is now well known as MAC (Model Algorithmic Controller).

The MPHC strategy relies on three principles [3].

1. The multivariable process to be controlled is represented by its impulse responses which constitutes the *internal model*. This model is used online for prediction, its inputs and outputs are updated according to actual state of the process. Though it could be identified online, this internal model is most of the time computed offline.

2. The strategy is fixed by means of a *reference trajectory* which defines the closed loop behavior of the plant. This trajectory is initiated in the actual output of the process and tends to the desired set-point.

3. Controls are not computed by a one shot operator or controller but through a procedure which is heuristic in the general case. Future inputs are computed in such a way that, when applied to the internal predictive model. they induce outputs as close as possible to the desired reference trajectory.

Control Algorithm

MAC generated an optimum control sequence $[u(k+1), \cdots, u(k+Hp)$. by minimizing the cost function of the form [1].

$$J = \sum_{i=1}^{H_p} [\hat{y}(k+i) - w(k+i)]^2 \gamma_i^2 \tag{2.8}$$

where w is the reference trajectory that specifies the desired response of the output. $\hat{y}$ is the predicted output and γ is the weighting matrix.

The different steps of the control algorithm are brought up in the diagram given in Figure 2.1 [3]. The A loop iterates to compute each predicted input vector sequence needed to obtain a fit between the internal model output and the reference trajectory for a number of sample times in the future.

Loop B iterates on the number of outputs to be controlled.

Loop C iterates on the whole predicted input control vector sequence ensuring convergence. Once this is satisfied only the first input control vector of the prediction horizon predicted sequences is applied.

Some basic theoretical properties of MAC or MPHC can be found in [33]

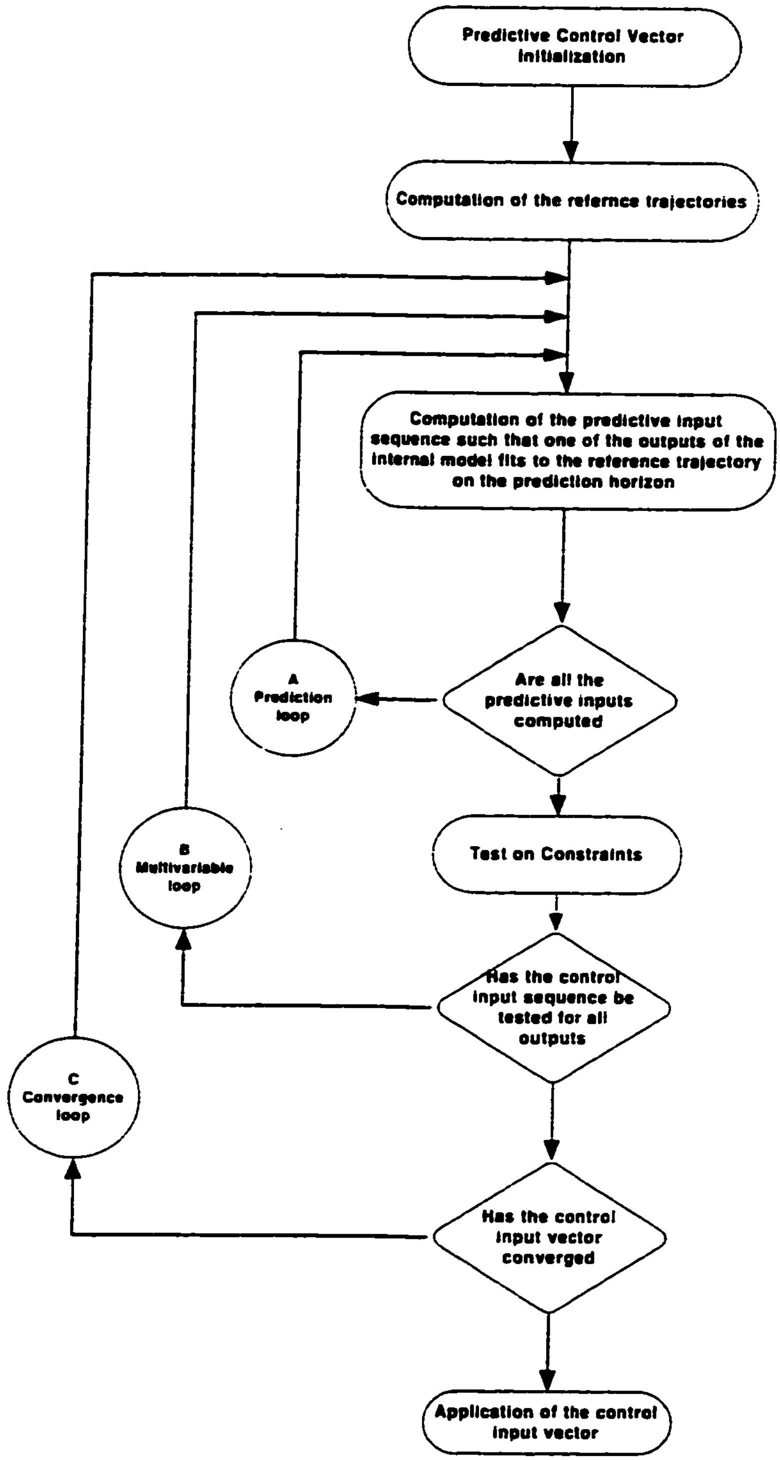

Figure 2.1: Control Flowchart of the MPHC Algorithm.

2.4.2 Dynamic Matrix Control(DMC)

Dynamic Matrix Control presented by Cutler et al. in 1979 uses a step response model of the process to predict the future behavior of the plant as contrast to MPHC which uses an impulse response model. In DMC, input moves are penalized. Thus the cost function in Equation 2.4 for the SISO case is now written as [1]

$$J = \sum_{i=1}^{H_p} [\hat{y}(k+i) - w(k+i)]^2 \gamma_i^2 + \sum_{i=1}^{H_c} [\Delta u(k+i-1)]^2 \beta_i^2 \tag{2.9}$$

where

$$\Delta u(k+i-1) = u(k+i) - u(k+i-1)$$

γ and β are the weighting matrices used in the tuning of the controller. The reason for adding the manipulated variables into the performance index is to prevent large swings in the manipulated variable inputs. The bigger the weighting factor β, the smaller the changes in the Δu's.

Some of the characteristics [15] of DMC are presented below.

- Only stable processes without integrators can be modelled and controlled by this controller.
- In DMC, steady state errors do not occur for constant setpoints and/or disturbances. However, nonconstant reference trajectories and/or disturbances can yield steady state errors.
- The design parameters of the DMC controller make it possible to realize mean-level and minimum settling time controllers.

- Eliminating certain frequencies in $u(k)$ is not possible.
- Good performance may require an excessive number of step response coefficients [34].

2.4.3 Generalized Predictive Controller

The GPC uses a Controlled Autoregressive and Integrated Moving Average **CARIMA** or simply transfer function model to predict the future behavior of the plant.

$$A(q^{-1})y(k) = B(q^{-1})u(k-1) + \frac{\varepsilon(k)}{\Delta} \tag{2.10}$$

where $A(q^{-1})$ and $B(q^{-1})$ are polynomials in the delay operator q^{-1}. $y(k)$ and $u(k)$ are output and control variables. respectively. and $\varepsilon(k)$ is an uncorrelated random sequence. Δ represents the differencing operator $1 - q^{-1}$.

It uses the same objective function as DMC given by Equation 2.9. Some of the Characteristics of GPC are presented below [15].

- Stable as well as unstable processes can be modelled.
- Nonconstant reference trajectories and/or disturbances can yield steady state errors.
- Eliminating certain frequencies in $u(k)$ is also not possible.

2.4.4 Other Predictive Controllers

Different predictive controllers are found in industry. Their underlying principles are the same, however. some differences are present. These differences are in the use of the model. in the objective function etc. For example, the PCA controller uses a finite impulse response model of the process like MAC but does not consider the control horizon. EPSAC controller considers the control horizon but usually it is kept constant at $H_c = 1$ [15], however, prediction horizon varies.

A good discussion on some well known commercially available predictive controllers can be found in [15].

2.5 Advantages and Disadvantages of MPC

Some of the advantages of MPC are [15]

- The concept of predictive control is not restricted to single-input, single-output processes. Predictive controllers can be derived for and applied to multi-input, multi-output processes.
- Predictive controllers can be applied to linear and nonlinear processes.
- Predictive control is the only methodology that can handle process constraints in a systematic way during the design of the controller.
- The concept of predictive control can be used to control a wide variety of processes without the designer having to take special precautions.

- In a natural way feed-forward action can be introduced for compensation of measurable disturbances and for tracking reference trajectories.

- Because predictive controllers make use of predictions, pre-scheduled reference trajectories (for example, used in robot control) or set points can be dealt with.

- Predictive control is an open methodology. That is, within the framework of predictive control there are many ways to design a predictive controller. As a result, different predictive controllers, each with different properties, have been proposed in the literature over the last decade. Some of them are mentioned in Chapter 1.

Some disadvantages are [15]

- Since predictive controllers belong to the class of model-based controller design methods, a model of the process must be available. In general, in designing a control system two phases can be distinguished: modelling and controller design. Predictive control provides only the solution for the controller design part. A model of the process must be obtained by other methods.

- A second drawback is due to the fact that the predictive control concept is an open methodology. It has already been mentioned that due to this, different predictive controllers can be derived having different properties. Although the differences between these controllers seem rather small, these small differences can yield very different behavior of the closed-loop systems. As a result, it can be quite difficult to select which predictive controller must or can be used to solve a particular control problem. One cannot afford the expense designing a control system that one knows will not work with another process and whose cost cannot therefore be spread over a large number of applications.

In the next chapter, process models are described in detail with emphasis on non-linear models. Different models used by the above mentioned predictive controllers are explained.

Chapter 3

Process Models

The model of the process is the heart of the Model Based Predictive Controller concept. All MPCs explicitly use a model of the process to be controlled to determine the future behavior or output of that plant or process. For an ideal MPC system, the model should match the process exactly. However, this is not the case in general. Different identification techniques have been employed for that purpose.

There are linear as well as non-linear models. In the general practice of linear MPC, the majority of dynamic models are derived from plant testing and system identification. Since most of the industrial plants are non-linear, they are emphasized in this thesis. However linear process models are also described here for completeness.

3.1 Linear Models

1. Impulse Response Model

2. Step Response Model

3. Transfer Function Model

4. State Variable Model

3.1.1 Impulse Response Model

The impulse response model can be defined mathematically as [15]

$$y(k) = \sum_{j=0}^{n_{H-1}} h_j u(k - j - 1) \tag{3.1}$$

where n_H is the number of impulse response elements h_j taken into account. All other elements $h_{n_H}, \cdots$ are assumed to be equal to zero. This model is called the *Finite Impulse Response Model.* This model has been implemented in Model Algorithmic Control (IDCOM) [3] and in Predictive Control Algorithm (PCA) [15]. The prediction of the process output at $t = k + i$ can be done in a simple manner with this model:

$$y(k+i) = \sum_{j=0}^{n_H - 1} h_j u(k - j + i - 1) \tag{3.2}$$

The main advantages of using a finite impulse response models are

- The prediction of the process output does not involve complex calculations.
- No assumptions has to be made about the order of the process.

A main disadvantage is that an FIR model many parameters to be known or estimate.

3.1.2 Step Response Model

The step response model can be defined mathematically as [15]

$$y(k) = \sum_{j=0}^{n_S-1} s_j \Delta u(k-j-1) \tag{3.3}$$

where Δ is the differencing operator ($\Delta = 1 - q^{-1}$) with q^{-1} the backward shift operator) and n_S is the number of step response elements s_j taken into account. All other elements $s_{n_s}, \cdots$ are assumed to be constant. This model is termed *Finite Step Response Model* (FSR) and has been implemented in Dynamic Matrix Control (DMC) [15].

The advantages and disadvantages of using a finite step response model are the same as the finite impulse response model.

3.1.3 Transfer Function Model

The transfer function model can be defined mathematically as [15]

$$y(k) = \frac{q^{-d}B(q^{-1})}{A(q^{-1})} u(k-1) \tag{3.4}$$

where d is the time delay of the process and the A and B are polynomials given by

$$A(q^{-1}) = 1 + a_1 q^{-1} + \cdots + a_{n_A} q^{-n_A}$$

$$B(q^{-1}) = b_o + b_1 q^{-1} + \cdots + b_{n_B} q^{-n_B}$$

where n_A and n_B are the degrees of the polynomials A and B. The prediction of the process output can be determined by the following relation using a transfer function model.

$$y(k+i) = \frac{q^{-d}B(q^{-1})}{A(q^{-1})}u(k+i-1) \tag{3.7}$$

Transfer function model has been implemented in GPC [35].

The advantages of a transfer function model includes

- a minimal number of parameters is required to describe a linear process.
- stable and unstable processes can be described by using the transfer function model.

The main drawbacks are

- an assumption about the order of the process must be made
- prediction of the process output described by a transfer function model is more complex than that of a process described by an FIR or FSR model.

3.1.4 State Variable Model

Every linear lumped system can be described by a set of equations of the form [36]

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{3.8}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{3.9}$$

for a system with p inputs, q outputs, and n state variable, **A**, **B**, **C**. and **D** are, respectively $n \times n, n \times p, q \times n$, and $q \times p$ constant matrices. Here x are the states, $\dot{x}$ represents the derivative of the state, u is the input and y is the output of the process. The I-step ahead prediction of the output of the process can be done using state space equations in a simple manner given by

$$\mathbf{x}(k+1|k) = \mathbf{A}\mathbf{x}(k|k) + \mathbf{B}\mathbf{u}(k|k) \tag{3.10}$$

$$\mathbf{y}(k+1|k) = \mathbf{C}\mathbf{x}(k|k) + \mathbf{D}\mathbf{u}(k|k) \tag{3.11}$$

where $x(k+1|k)$ means the prediction of x at time $k+1$ given the information at time k.

3.2 Non-linear Models

Since the focus in this thesis is on nonlinear processes. therefore they are explored in detail. This section describes some common and well known nonlinear models which are used in predictive control. some of them are

1. Hammerstein Model
2. Wiener Model
3. NARMAX (Nonlinear Auto Regressive and Moving Average with Exogeneous Inputs)
4. Nonlinear State Variable Model

These are explored in the following section.

3.2.1 Hammerstein Model

The Hammerstein model has a special structure that facilitates the analysis of nonlinear processes. Many practical examples that are modelled by a Hammerstein model can be found. These include: the distillation column [21], heat exchanger [37], pH neutralization processes [21], furnaces and reactors. It consists of a linear dynamic element followed by a nonlinear zero-memory block. The Hammerstein model of a nonlinear system is shown in Figure 3.1.

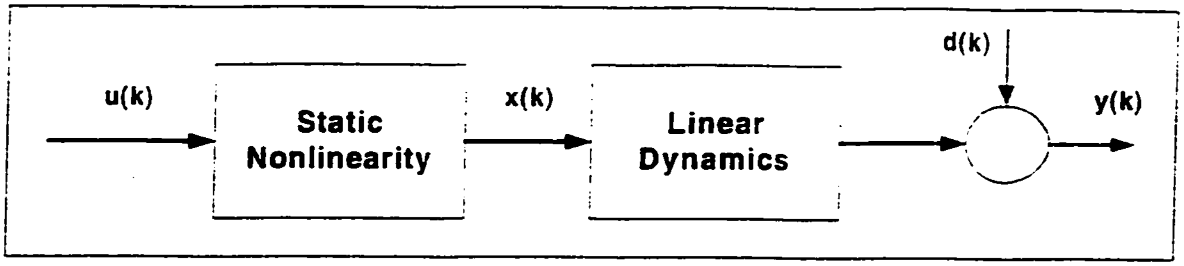

Figure 3.1: Hammerstein Model.

The static NL element scales the input $u(k)$ and transforms it to $x(k)$, and the dynamics are modeled by a linear discrete transfer function $G(z^{-1})$, whose output is $y(k)$. The advantage of a Hammerstein model is that, it effectively isolates the nonlinearity from the linear dynamics keeping the inherent advantages of a linear MPC. The Hammerstein model models the NL effects as an input-dependent gain nonlinearity. The slope of the nonlinearity at a certain operating point is the instantaneous gain of the system. If the static NL system function is assumed to be approximated by a finite polynomial expansion, the Hammerstein model can be described by the following equations [37].

$$
\begin{aligned}
y(k) + a_1 y(k-1) &+ \cdots + a_n y(k-n) \\
&= b_1 x(k-1) + b_2 x(k-2) + \cdots + b_n x(k-n) \qquad (3.12) \\
x(k) &= N(u(k)) \qquad (3.13)
\end{aligned}
$$

where $u(k)$ is the input to the system. $y(k)$ is the output of the system. N is the nonlinearity and $x(t)$. the output of the nonlinear block is the intermediate or state variable. The nonlinearity can be separately parameterized as shown in Figures 3.2 and 3.3 for a 2×2 system. The commonly parameterized nonlinearity is the more general case given by Equation 3.14 for a two-input two-output system.

$$x_1(k) = N_1(u_1(k), u_2(k)) \qquad (3.14)$$

$$x_2(k) = N_2(u_1(k), u_2(k)) \qquad (3.15)$$

Simulations examples for both types of nonlinearities are provided in Chapter 4. Equation 3.12 can be written in the following form in which the intermediate variable $x(k)$ has been removed.

$$y(t) = \frac{B(q^{-1})}{A(q^{-1})} \sum_{i=1}^{m} \gamma_i u^i(k) \qquad (3.16)$$

where the polynomial $A(q^{-1})$ and $B(q^{-1})$ are

$$A(q^{-1}) = 1 + a_1 q^{-1} + \cdots + a_n q^{-n}$$

$$B(q^{-1}) = b_1 q^{-1} + \cdots + b_n q^{-n}$$

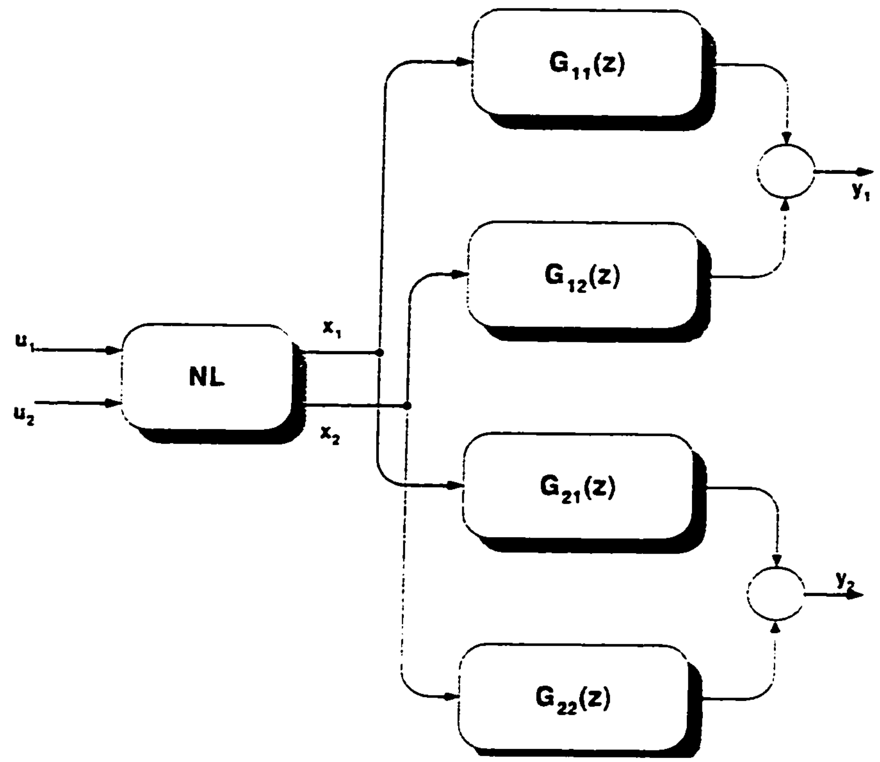

Figure 3.2: Commonly Parameterized Hammerstein Model for a two-input, two-output system.

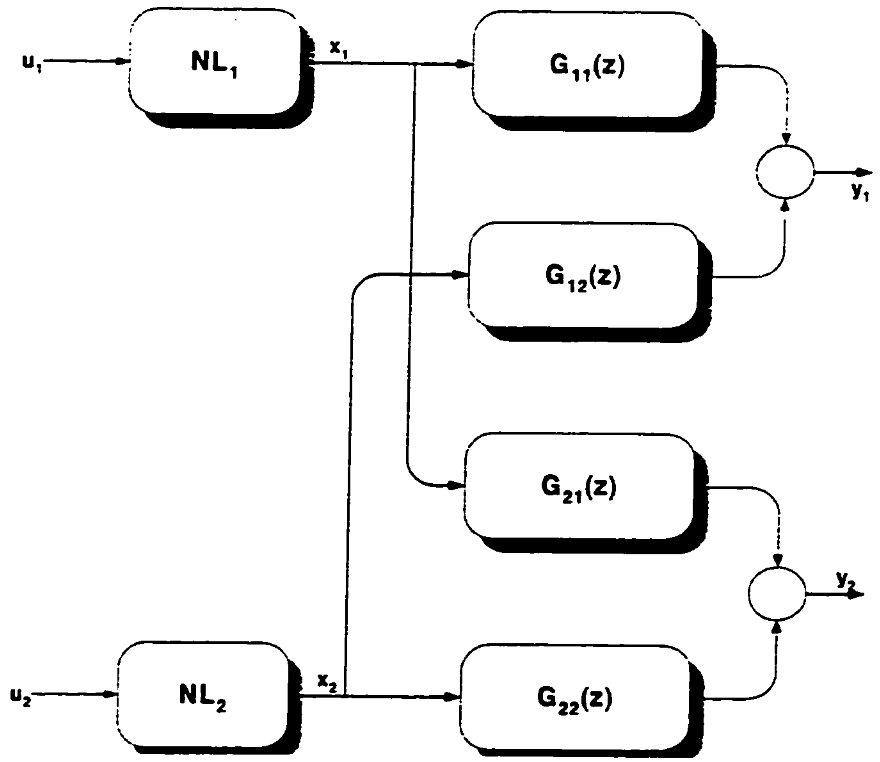

Figure 3.3: Seperately Parameterized Hammerstein Model for a two-input, two-output system.

3.2.2 Wiener Model

The *Wiener model* of a nonlinear systems is the model which is constructed by a nonlinear gain cascaded after a linear subsystem or in other words. it is the transpose of a Hammerstein Model. The choice of the nonlinear element is virtually unlimited. For the linear element, a variety of options are also available including the step response and the transfer function-based models. Many practical examples are found based on Wiener models. Some of them are Control Valve [38]. pH Neutralization [19] and an Industrial C-2 Splitter [18]. The structure of Wiener model of a nonlinear system is shown is Figure 3.4.

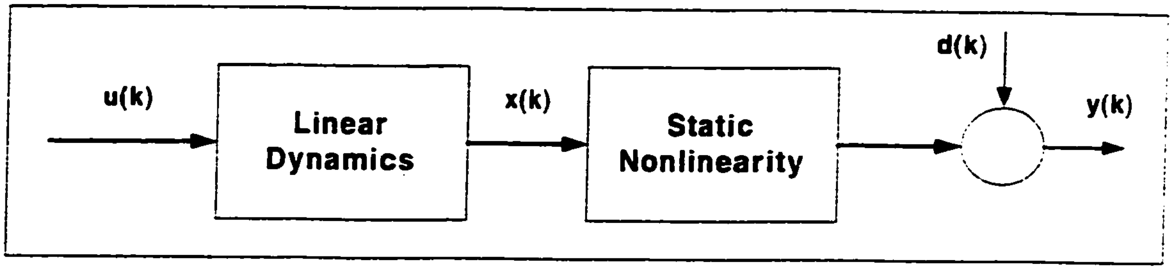

Figure 3.4: Wiener Model.

where $y(k)$ is the output of the nonlinearity given by

$$y(k) = N(x(k)) \tag{3.19}$$

The linear subsystem of the Wiener model is presented by the following ARMAX model:

$$x(k) = b_1u(k-1) + \cdots + b_{nb}u(k-nb) \tag{3.20}$$

$$+ \; a_1x(k-1) + \cdots + a_{na}x(k-na) \tag{3.21}$$

or in transfer function form as

$$x(k) = \frac{B(q^{-1})}{A(q^{-1})}u(k) \tag{3.22}$$

where $u(k)$ is the input to the process and $x(k)$ is the state variable.

The Wiener model for the case of a two-input, two-output system is shown in Figure 3.6.

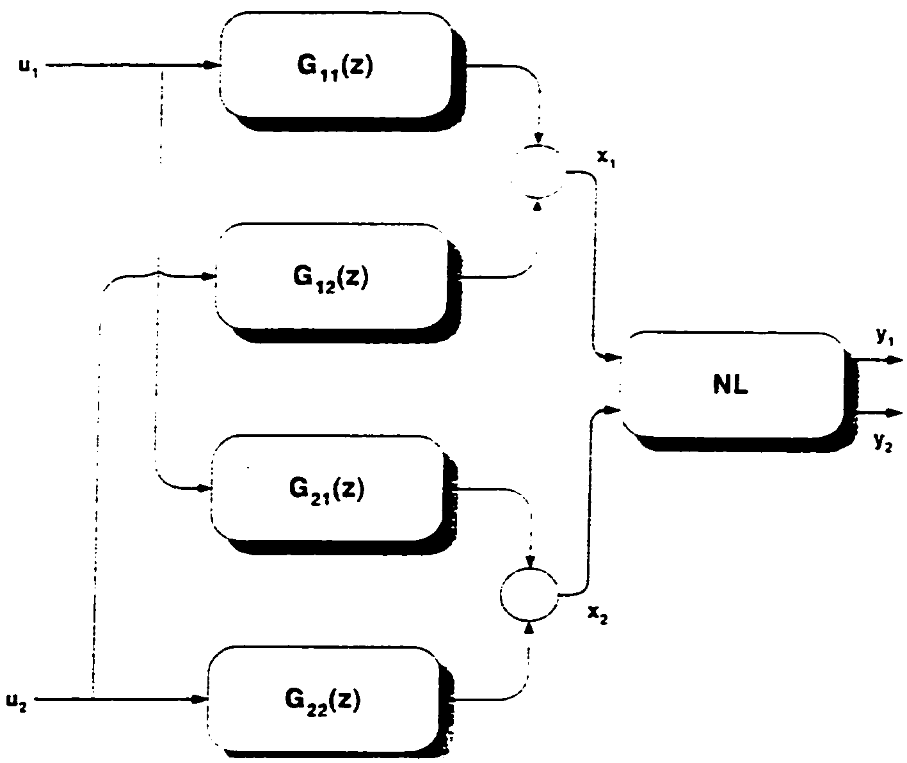

Figure 3.5: Commonly Parameterized Wiener Model for a two-input, two-output system.

3.2.3 NARMAX Model

An Nth-order **NARMAX** model can in general be expressed by

$$y(n) = F(y(n-1), \cdots, y(n-N), u(n), u(n-1), \cdots, u(n-N)) \tag{3.23}$$

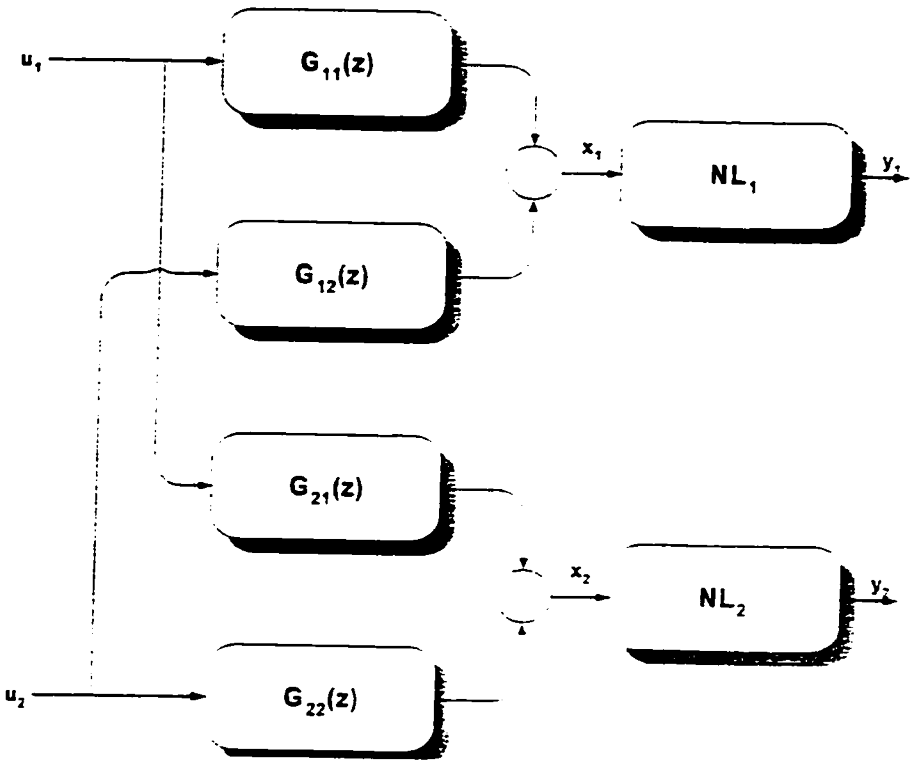

Figure 3.6: Seperately Parameterized Wiener Model for a 2-input 2-output system.

where $F(\cdot)$ is some nonlinear function. $y(n)$ and $u(n)$ are the current output and input, and $y(n-i)$'s and $u(n-i)$'s for $i \neq 0$ are the delayed output and inputs. As of wiener and hammerstein models, identification of NARMAX also consists of two mutually independent parts, the linear part and the nonlinear part. The nonlinear characteristics of these models are often assumed to be of polynomial form, and usually with memory. For identification of the nonlinear part, some regression method is usually employed.

3.2.4 Nonlinear State Variable Model

Most physical systems are nonlinear and time varying. Some of them can be described by the nonlinear differential equation of the form

$$\mathbf{x}(k+1|k) = \mathbf{h}(\mathbf{x}(k|k), \mathbf{u}(k|k), k) \quad (3.24)$$

$$\mathbf{y}(k+1|k) = \mathbf{f}(\mathbf{x}(k|k), \mathbf{u}(k|k), k) \quad (3.25)$$

where $\mathbf{h}$ and $\mathbf{f}$ are nonlinear functions. The behavior of such equations can be very complicated.

3.3 Disturbance Models

In order to predict the output of the process the disturbances must also be considered and modeled as they cannot be ignored in practice. In order to take these disturbances into account when predicting the output of the process, the disturbance must also be modelled. For this purpose the models which were defined before can be extended to include the effect of the disturbance. Since disturbances are mostly additive in nature an extra term must be added to the model of the process. Lets take the case of Transfer function model. If $n(k)$ is the disturbance then this term can be included in the model in the following way.

$$y(k) = \frac{q^{-d}B(q^{-1})}{A(q^{-1})}u(k-1) + n(k) \quad (3.26)$$

Disturbances can be classified into the following categories.

1. Deterministic disturbances

2. Stochastic Disturbances

where the deterministic disturbance is a disturbance which can be measured. while the stochastic disturbance is a dicrete white noise sequence with zero mean and variance σ^2. Some disturbance models can be found in [15].

Chapter 4

GA Based Model Predictive Control

The proposed genetic-based control algorithm is shown in Figure 4.1. The GA-based controller uses the process model to search for the control moves, which satisfies the process constraints and optimizes some cost function. The process model used could be of any type. The algorithm is described by the flowchart in Figure 4.2, and is explained in the following section.

4.1 Problem Formulation

The proposed algorithm mainly consists of applying Genetic Algorithms to the optimization phase of nonlinear model predictive controller. The algorithm starts by generating random numbers in some desired range. These random numbers form

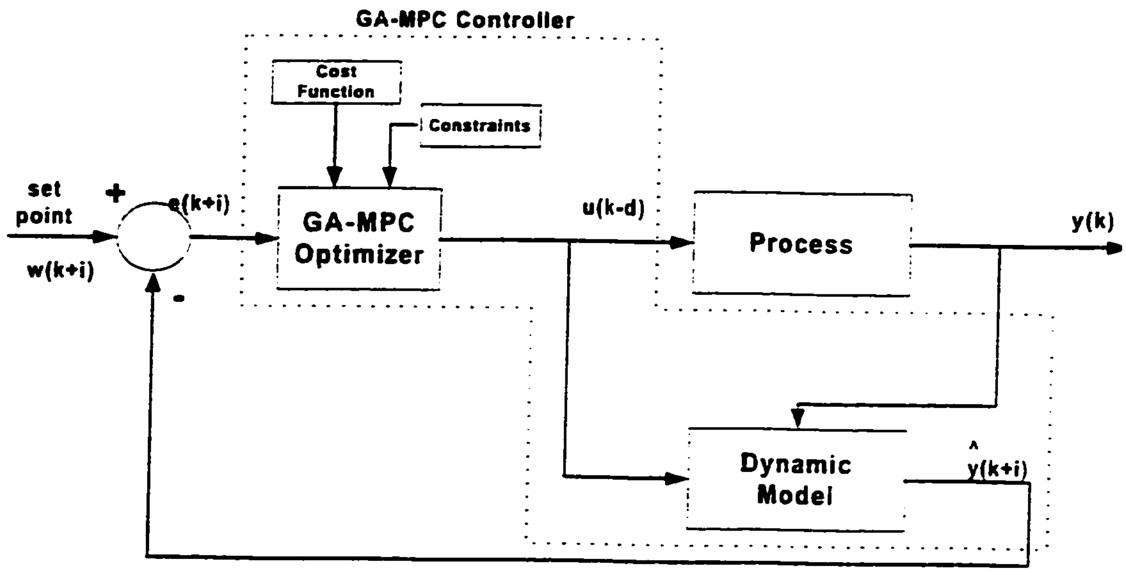

Figure 4.1: Proposed GA-based MPC.

the initial population of the GA and also the initial inputs to the process. Initially we don't have any information about the process so the population must be random. The set of inputs generated are applied to the model of the process and an objective function is evaluated from which a fitness function is evaluated. The fitness function used throughout this thesis is given by

$$fitness = \frac{1}{J+1} \tag{4.1}$$

where J is the performance index or the cost function.

The GA then manipulates these inputs according to the objective function values and finds the optimal inputs or control moves for the process. The population taken consists of real valued chromosomes. The real coded GA in this case has two advantages. First, it does not require a conversion from one base to another, thus saving computation time and second is that, most of the real world problems deal

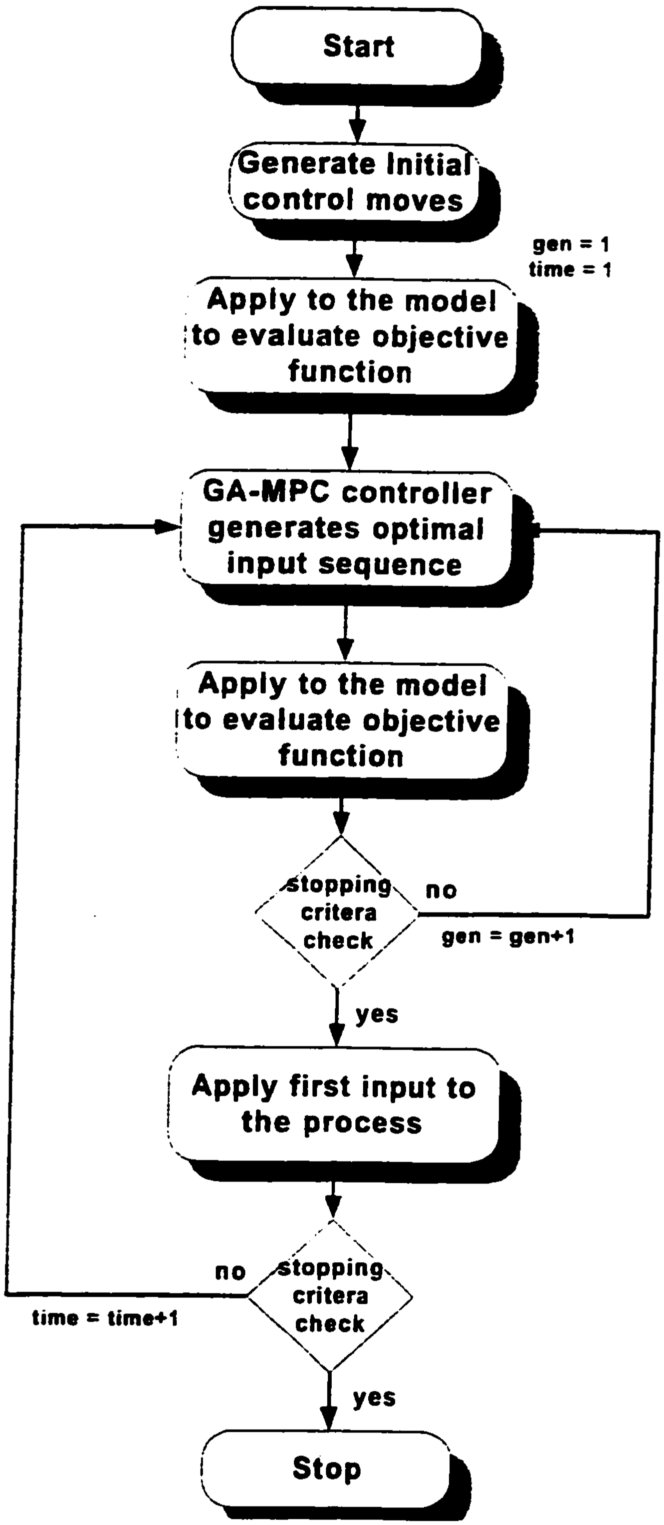

Figure 4.2: Flow chart of the proposed GA-based MPC.

with real numbers so the simulations will give us much better view than what we have if we use binary numbers or integers. The length of the chromosome represents the length of the prediction horizon (H_p). If there are more than one inputs then the length of the chromosome will be doubled for every input added to the system. The performance index or objective function evaluated usually is the sum of the square of the errors between actual and predicted outputs over a finite prediction horizon (H_p) augmented with the weighted sum of the square of the change in inputs over the control horizon and the weighted sum of the square of the input moves over the prediction horizon. The goal is to minimize this objective function which is given by Equation 4.2.

$$J = \sum_{i=1}^{H_p} e(k+i)^T Q e(k+i) + \sum_{i=1}^{H_c} \Delta u(k+i)^T R \Delta u(k+i) + \sum_{i=1}^{H_p} u(k+i)^T S u(k+i) \qquad (4.2)$$

subject to

$$u^l \leq u(k+i) \leq u^u$$

$$\Delta u^l \leq \Delta u(k+i) \leq \Delta u^u$$

$$y^l \leq y(k+i) \leq y^u$$

where the superscripts l and u represents the lower and upper bounds respectively. Q is the weight on the prediction error

$$e(k) = \hat{y}(k) - w(k)$$

where w(k) is the reference or the desired setpint. R and S are weights on the change in the input Δu and magnitude of the input u.

Next the three genetic operators (selection, crossover and mutation) are applied to the inputs. The selection is performed on the basis of the fitness value from the objective function. Solutions having larger fitness value or lower objective function value have a higher chance to survive for the next generation.

The inputs now obtained are again applied to the process and new response is evaluated by using the model of the process. Again the objective function is evaluated over the prediction horizon and is given to the GA-based controller which again manipulates it and generates a new input sequence. This process is repeated upto a pre-specified number of generations. The sequence of inputs obtained at the end of this generational loop are thus optimal.

Now only the first input of the sequence is applied to the process discarding the rest of them. The advantage of GA is that multiple solutions are available from which the best solution which optimized the objective function and satisfy the process constraints can be selected.

The optimized or optimal input is thus applied to the process and new or updated response is obtained. This updated response is fedback to the GA-based controller which now evaluate the next input sequence by taking the feedback output as its initial value or starting point.

Constraints Formulation

The constraints are placed on inputs. and rate of change of inputs. The input constraints are implemented by generating initial population. which is usually random. in the desired range.

For the implementation of rate of change of input constraint. the input for current time instant. k is compared with the input of the previous time instant. $k-1$. If the difference Δu of the two inputs is violating the constraint. i.e.. if it is higher or lower than the desired range, it is set to the extreme value accordingly by manipulating the input at the current time instant. k. Thus not allowing the rate of change of inputs to violate the constraint.

4.2 Simulations

In this section. simulation results are presented. Results for the case of nonlinear SISO and MIMO systems are shown with different process models. Results for the case of linear SISO and MIMO systems are also shown for the completeness of the algorithm. All simulations were done in MATLAB running on a 500 MHz Pentium-III machine with extensive use of different toolboxes.

An example involving constraints on rate of change of input is considered. Model uncertainty is dealt next. Simulation results are presented for the case of model/plant mismatch and a solution is proposed for this problem. The proposed solution is called *GA-Based Adaptive Predictive Conrtrol.*

Some noisy environment measurements are discussed then. In the last section, tuning parameters of the proposed algorithm are discussed which are mainly prediction and control horizons, population size and number of generations in a time instant. Some comparisons and simulation results are also presented.

4.2.1 GA-MPC of a Linear SISO Process

In this section the proposed algorithm is applied to a Linear system. The process taken is described by the following transfer function.

$$G(q^{-1}) = \frac{0.0712q^{-1} + 0.0639q^{-2}}{1 - 1.591q^{-1} + 0.726q^{-2}} \tag{4.3}$$

This is a simple second order linear system. The block diagram of the proposed control scheme for the case of a linear process is shown in Figure 4.3 The model is

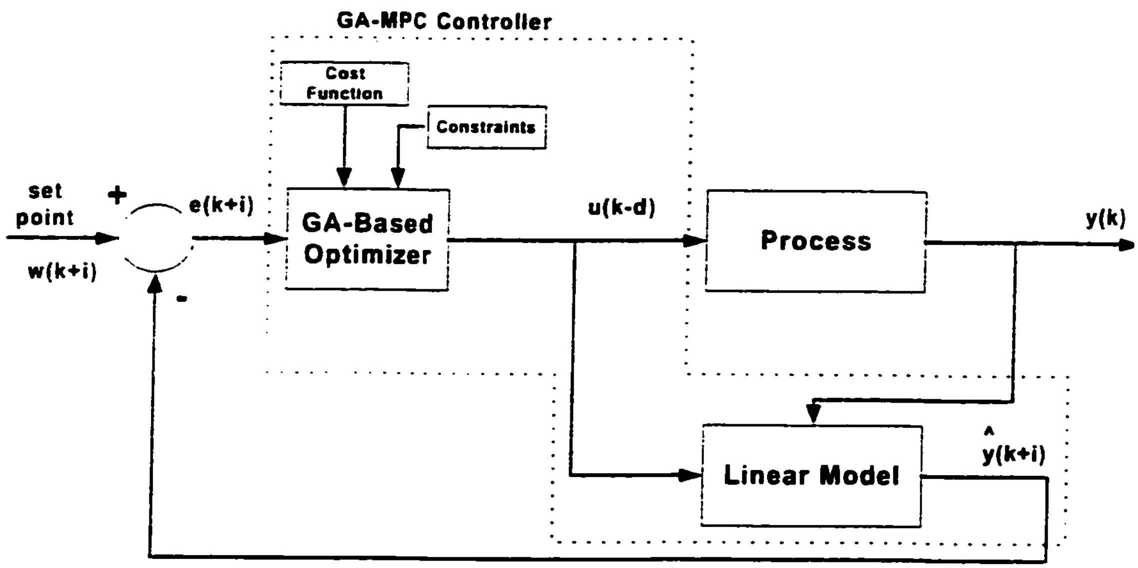

Figure 4.3: Proposed GA-based MPC for a Linear Process.

assumed to be identical to the process. The parameters used in the simulation are given in Table 4.1.

	Population size	Number of Generations	Mutation Probability	Crossover Probability	Prediction Horizon H_p	Control Horizon H_c
Tracking	100	30	0.005	0.7	10	1
Servo	100	30	0.005	0.7	10	1

Table 4.1: Parameters used in the simulation of a Linear SISO Process.

Two cases are discussed here. The first case is for a constant reference i.e.. the tracking performance of the algorithm. The second case is for step changes in set point or the servo performance. Input constraints for both cases are taken as

$$-1 \leq u \leq 1 \tag{4.4}$$

For the first case. the set point was taken as 0.7 and for the second case. it was taken as [0.8. 0.4. 0.8. 0.4] with step changes at every 100 sec. The weights used in the objective function are 1, 0 and 0 for Q, R and S respectively. The proposed algorithm was applied to the system and the response of the system is shown in Figure 4.4 and Figure 4.6 for constant reference and change of reference respectively.

The optimal inputs generated by the controller are shown for the two cases in Figures 4.5 and 4.7. The output response shows good tracking behavior. There is some overshoot, however, which can be minimized by proper selection of the tuning parameters or limit the rate of change of input by imposing constraints on it.

For the set point change case, the behavior is quite remarkable as the controller quickly adjusts itself for a change in desired operating point as seen in Figure 4.7.

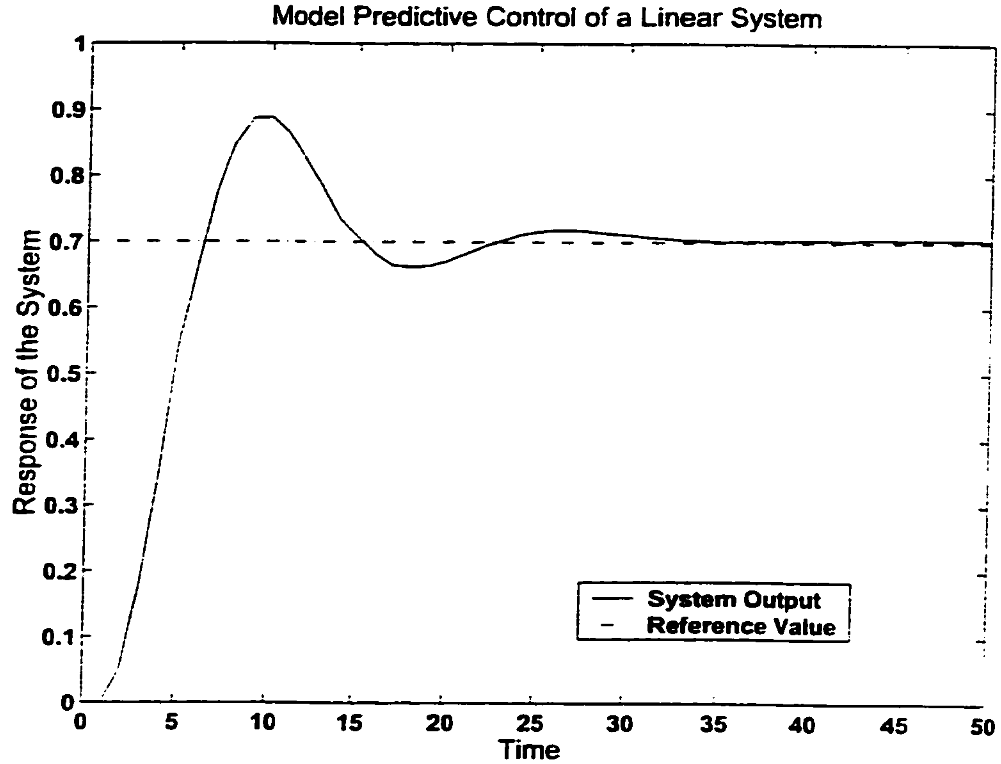

Figure 4.4: Application of the proposed algorithm to a Linear Process (Tracking Performance).

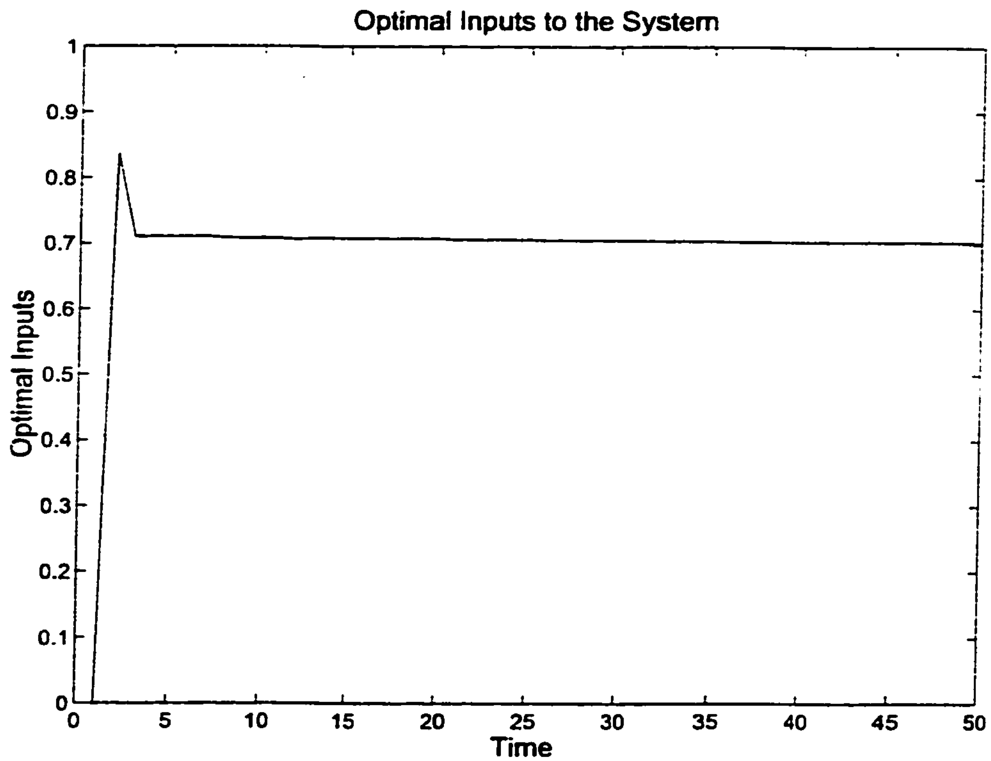

Figure 4.5: Optimal Inputs generated by the controller for a Linear Process (Tracking Case).

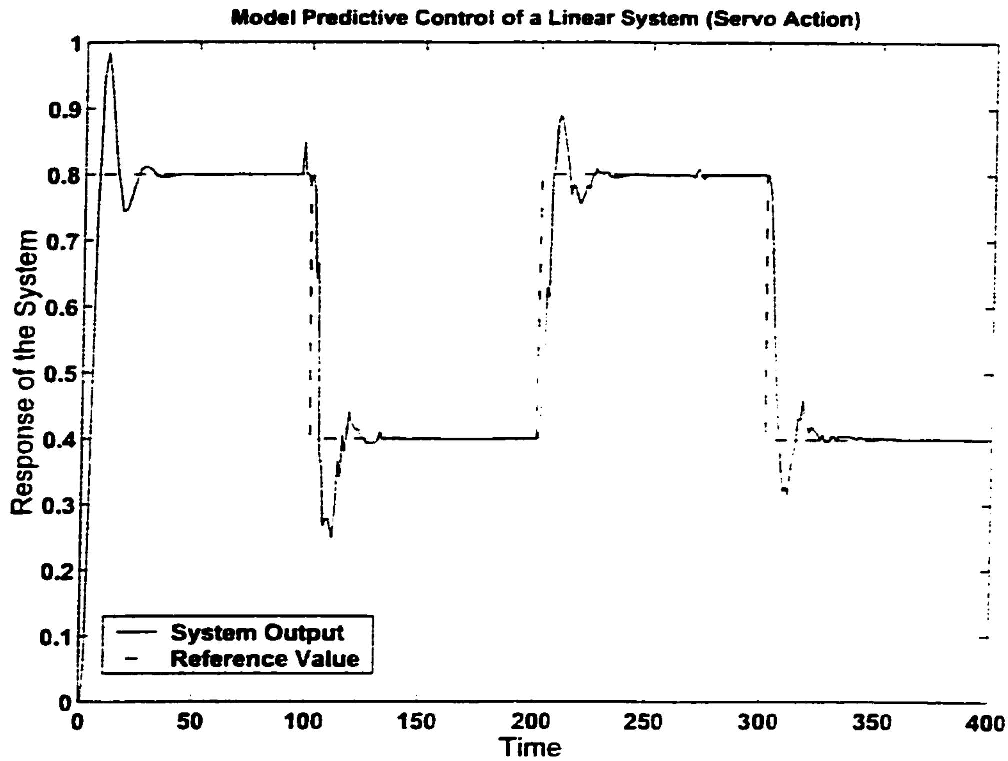

Figure 4.6: Application of the proposed algorithm to a Linear Process (Servo Performance).

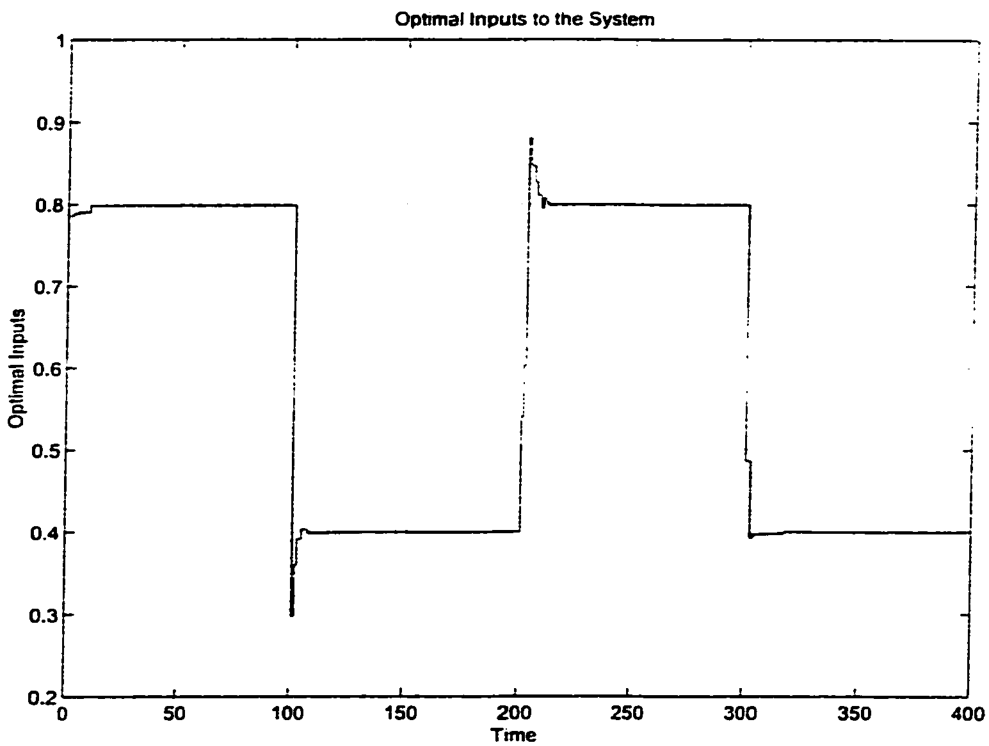

Figure 4.7: Optimal Inputs generated by the controller for a Linear Process (Servo Case).

4.2.2 GA-MPC of Nonlinear SISO Processes

The proposed algorithm was applied to two different types of nonlinear systems. The first one is a control valve which is described as a Wiener Model in [38]. The second is a heat exchanger which is described as a Hammerstein model in [37]. These are presented below.

Control Valve

The control valve is used to control fluid flow. It is simply an opening with adjustable area. Normally it consists of an actuator, a valve body and a valve plug. The actuator is a device that transforms the control signal to movement of the stem and valve plug. The model was obtained from [38], which describes the control valve as a wiener model. Wiener model is a special type of nonlinear model which consists of a linear dynamic block followed by a nonlinear zero memory block. Wiener model has been described in detail Chapter 3. The model is described by the linear dynamics in Equation 4.5 and the nonlinearity in Equation 4.6.

$$x(k) = \frac{0.0616q^{-1} + 0.0543q^{-2}}{1 - 1.5714q^{-1} + 0.6873q^{-2}} u(k) \tag{4.5}$$

$$y(k) = \frac{x(k)}{\sqrt{0.10 + 0.90x^2(k)}} \tag{4.6}$$

where $u(k)$ is the control pressure, $x(k)$ is the stem position, and $y(k)$ is the flow through the valve which is the variable of interest. The nonlinear characteristic of the control valve is shown in Figure 4.8.

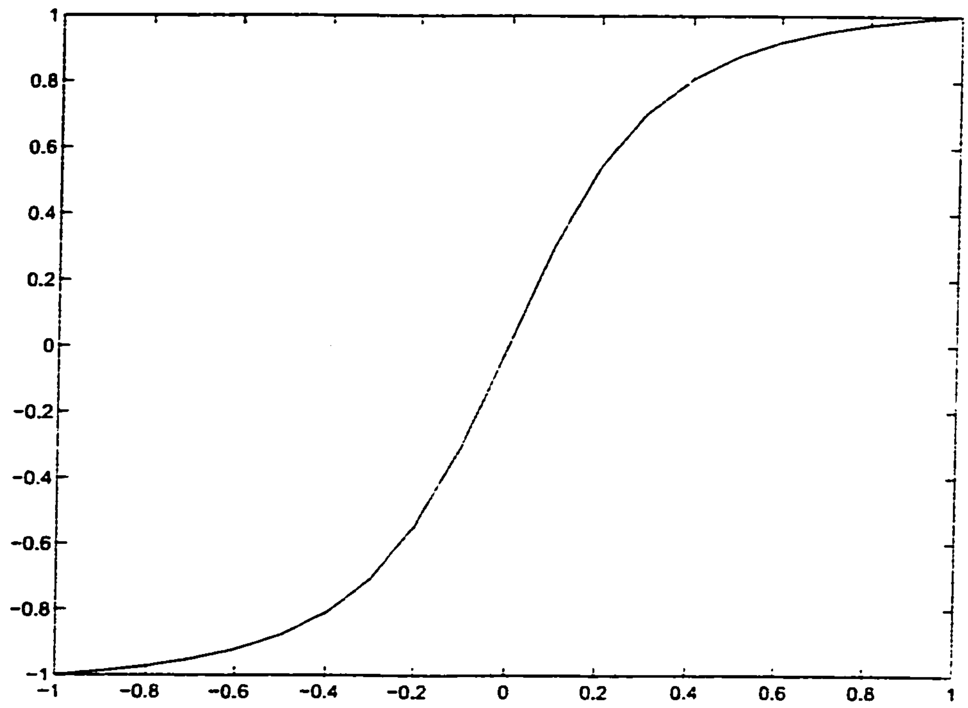

Figure 4.8: Nonlinear characteristics of a Control Valve.

The proposed control strategy for a wiener model is shown in Figure 4.9. It is clear that for this scheme to be applicable, the inverse of the nonlinearity should exist, otherwise the system response will go unbounded.

The parameters used for the simulation are provided in Table 4.2.

	Population size	Number of Generations	Mutation Probability	Crossover Probability	Prediction Horizon(H_p)	Control Horizon(H_c)
Tracking	100	30	0.005	0.7	10	1
Servo	100	100	0.005	0.7	3	3

Table 4.2: Parameters used in the simulation of a Control Valve.

The weights Q, R and S are taken as 1, 0.02 and 0. The control objective is to keep the process output as close as possible to the reference.

From the nonlinear characteristic curve, one can conclude that the input should not

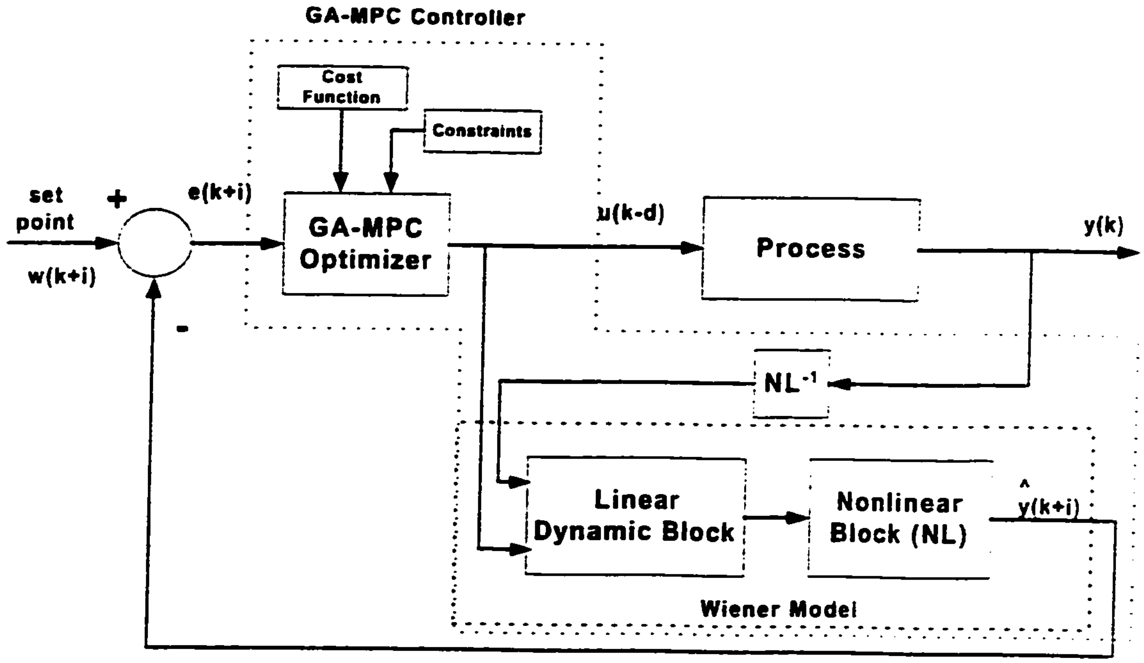

Figure 4.9: Block diagram of the proposed control strategy for Wiener Model.

exceed the saturation limit of the control valve. The saturation limit of the control valve is -1 on the lower side and +1 on the upper side, therefore the inputs for this system must be within this range. The constraints on input for both cases(tracking and set point change) is taken as

$$0 \leq u \leq 1$$

The proposed genetic-based control algorithm was simulated and the results are shown in Figure 4.10 which clearly demonstrates the successful performance of the proposed control algorithm. The performance of the proposed algorithm for different set points can be seen in Figure 4.12. The parameters used for this case are also provided in Table 4.2 and the weights are 1, 0 and 0 for Q, R and S respectively. Again the performance of the proposed algorithm is excellent in tracking different

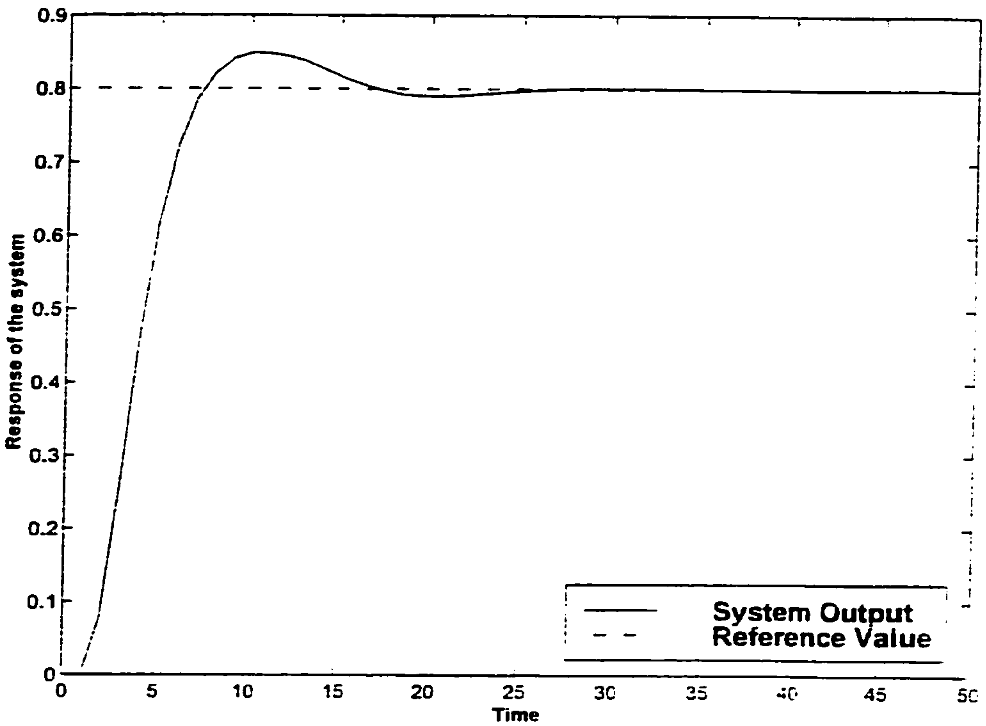

Figure 4.10: Application of the proposed algorithm to Control Valve (Tracking Performance).

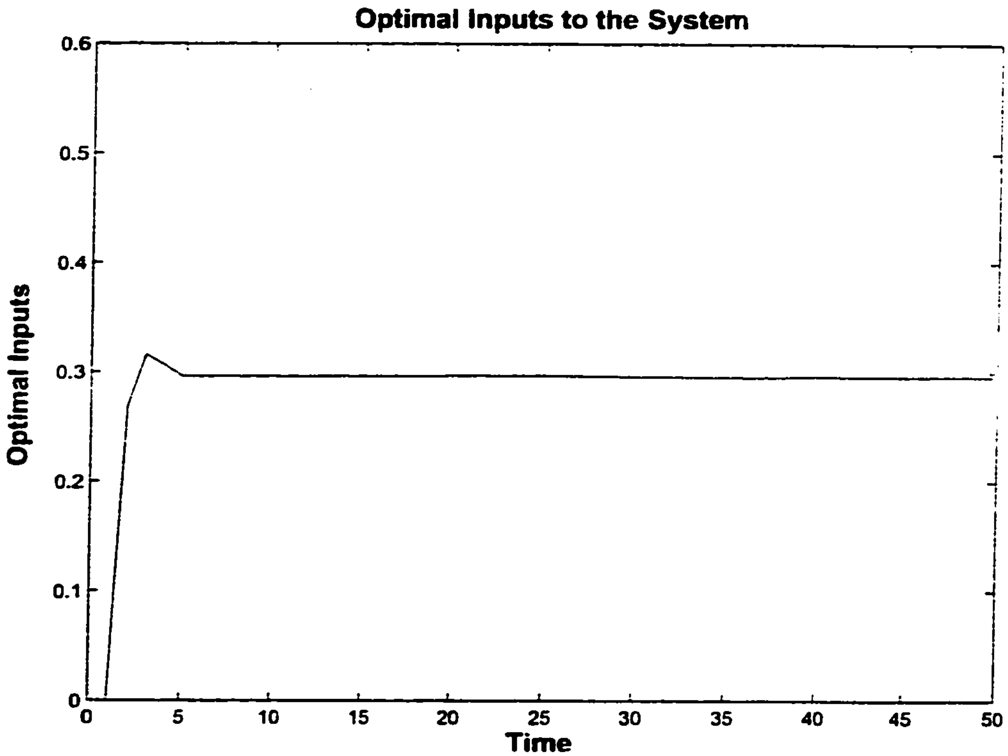

Figure 4.11: Optimal Inputs generated by the controller for the control valve (Tracking Case).

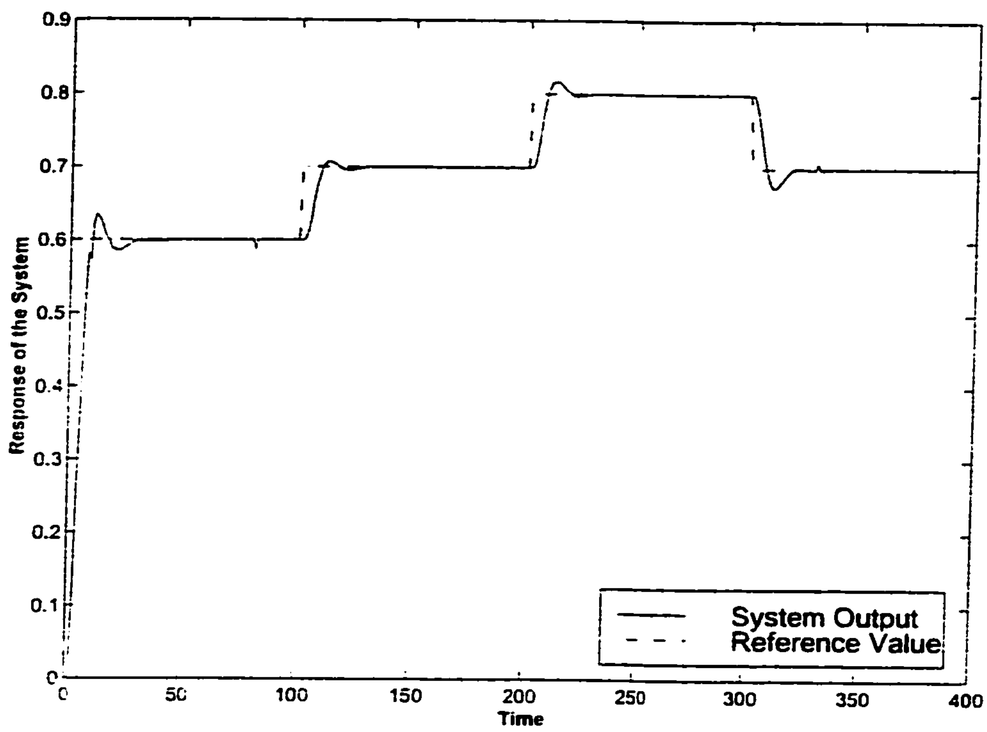

Figure 4.12: Application of the proposed algorithm to Control Valve (Servo Performance).

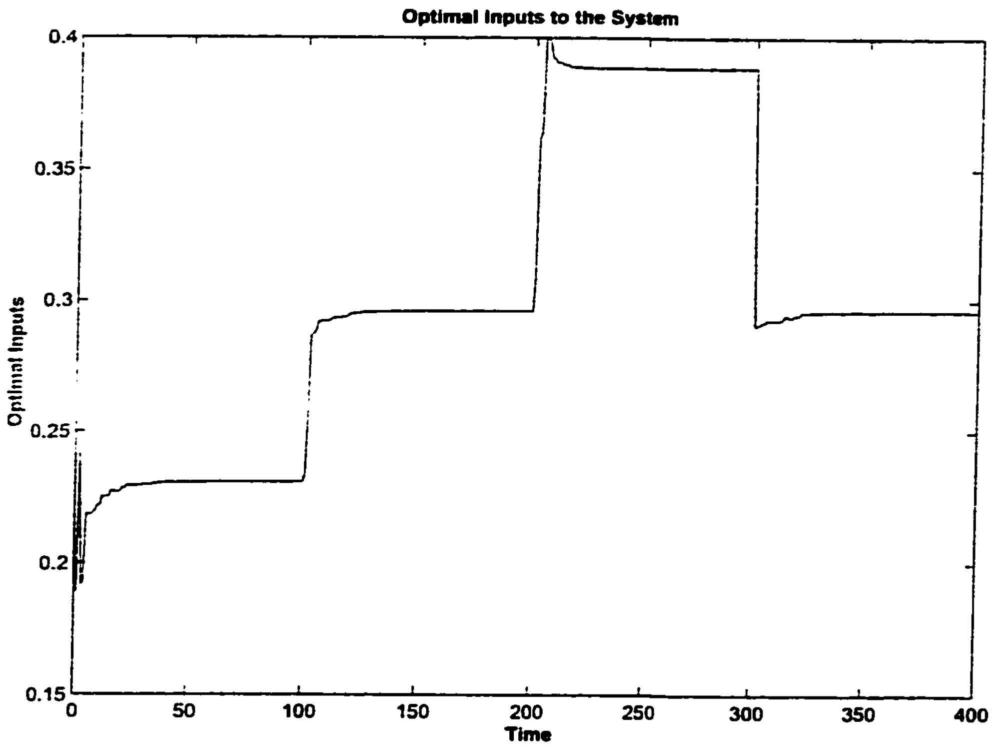

Figure 4.13: Optimal Inputs generated by the controller for the control Valve (Servo Case).

operating points. The inputs are converging to their optimal values which can easily be verified from the nonlinear characteristics. The optimal inputs generated for the two cases are shown in Figures 4.11 and 4.13.

Heat Exchanger

In this section, the proposed algorithm is applied to a heat exchanger. The process flow rate is the manipulated variable and the process exit temperature is the controlled variable. It is desirable to keep the exit temperature at a constant level. The model for the heat exchanger problem was taken from [37] which describes the heat exchanger as a *Hammerstein Model* given by

$$x(k) = -31.549u + 41.732u^2 - 24.201u^3 + 68.634u^4 \tag{4.7}$$

$$y(k) = \frac{0.207q^{-1} - 0.1764q^{-2}}{1 - 1.608q^{-1} + 0.6385q^{-2}} x(k) \tag{4.8}$$

where $x(k)$ is the nonlinearity, u is the process flow rate and $y(k)$ is the process exit temperature. The nonlinear characteristics of the heat exchanger is shown in Figure 4.15.

The parameters for this simulation are provided in Table 4.3 and the weights Q, R and S used in the objective function are $1, 0$ and 0 respectively. The constraints on the inputs are imposed as

$$0 \leq u \leq 1$$

The proposed genetic-based control algorithm was simulated and the results are

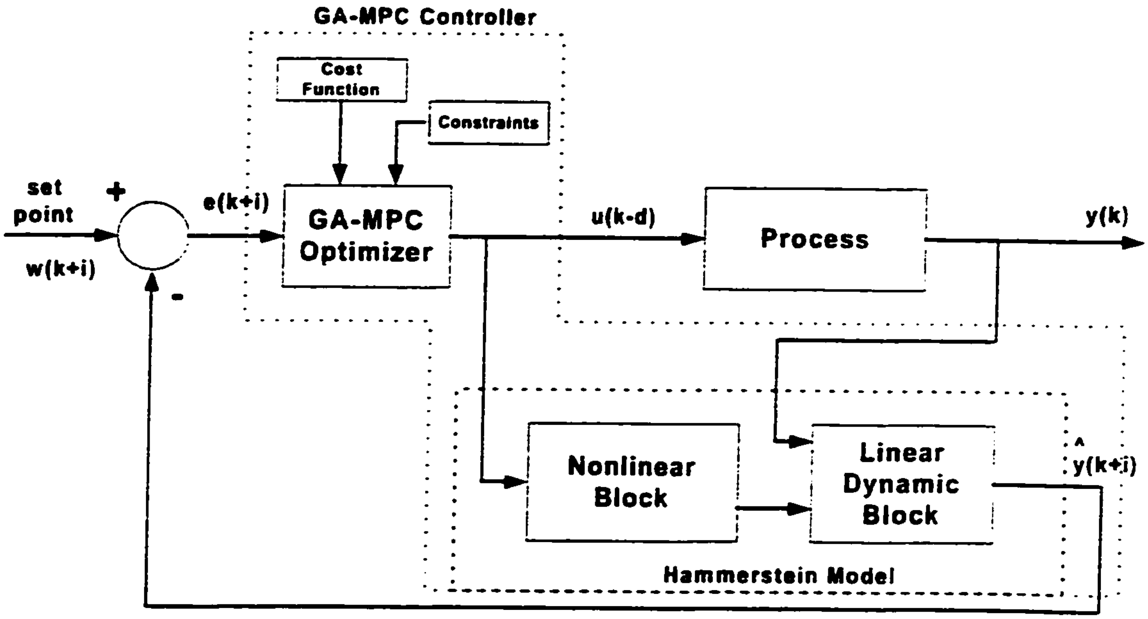

Figure 4.14: Block diagram of the proposed control strategy for Hammerstein Model.

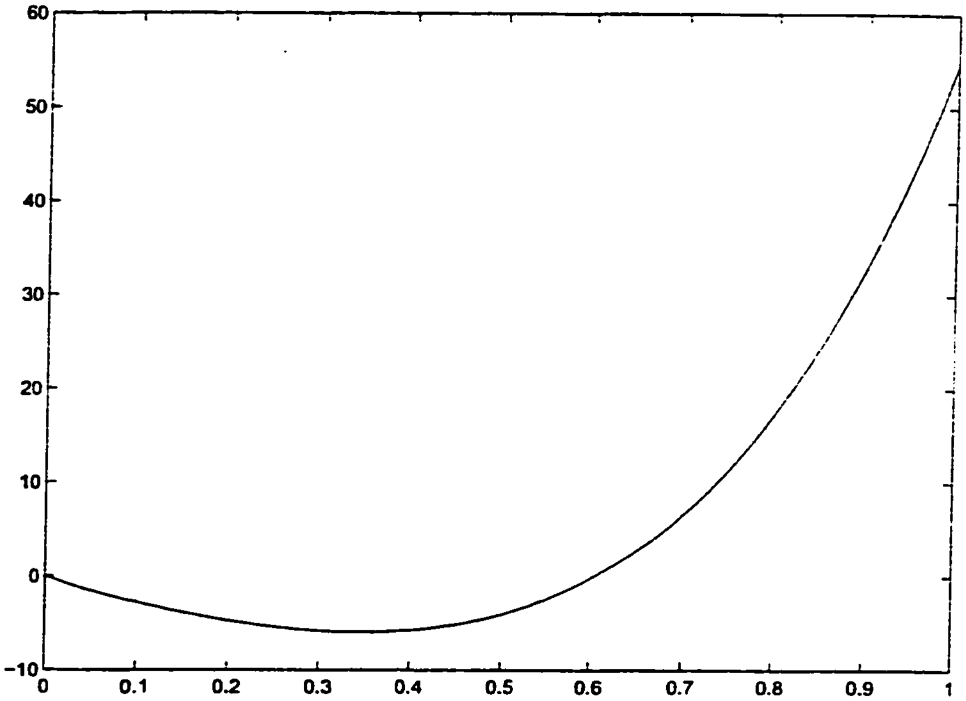

Figure 4.15: Non-linear characteristics of a Heat Exchanger.

	Population size	Number of Generations	Mutation Probability	Crossover Probability	Prediction Horizon(H_p)	Control Horizon(H_c)
Tracking	50	400	0.005	0.7	10	1

Table 4.3: Parameters used in the simulation of a Heat Exchanger.

shown in Figure 4.16 which clearly demonstrates the successful performance of the proposed control algorithm. The optimal inputs generated by the controller is shown in Figure 4.17.

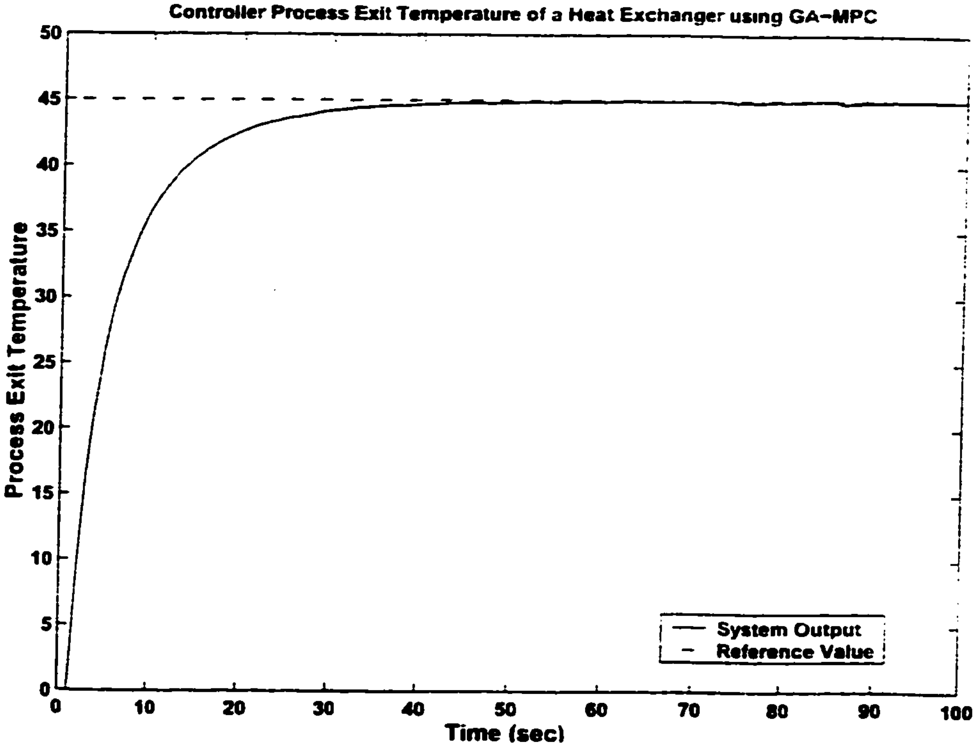

Figure 4.16: Application of the proposed algorithm to Heat Exchanger .

	Population size	Number of Generations	Mutation Probability	Crossover Probability	Prediction Horizon(H_p)	Control Horizon(H_c)
Tracking	50	400	0.005	0.7	10	1

Table 4.3: Parameters used in the simulation of a Heat Exchanger.

shown in Figure 4.16 which clearly demonstrates the successful performance of the proposed control algorithm. The optimal inputs generated by the controller is shown in Figure 4.17.

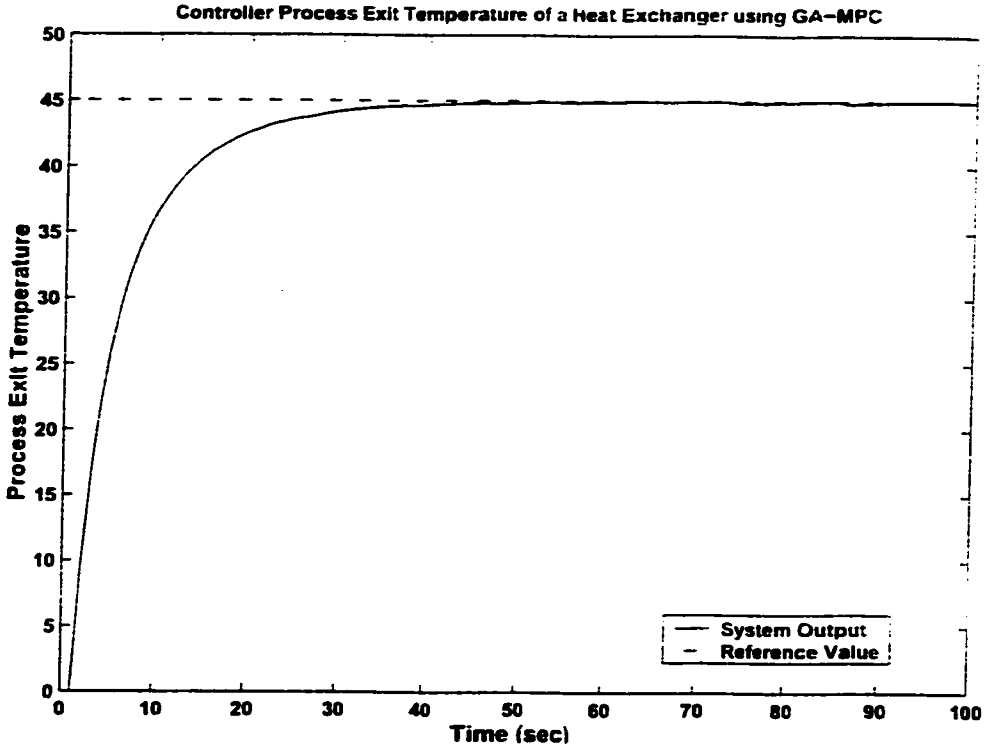

Figure 4.16: Application of the proposed algorithm to Heat Exchanger .

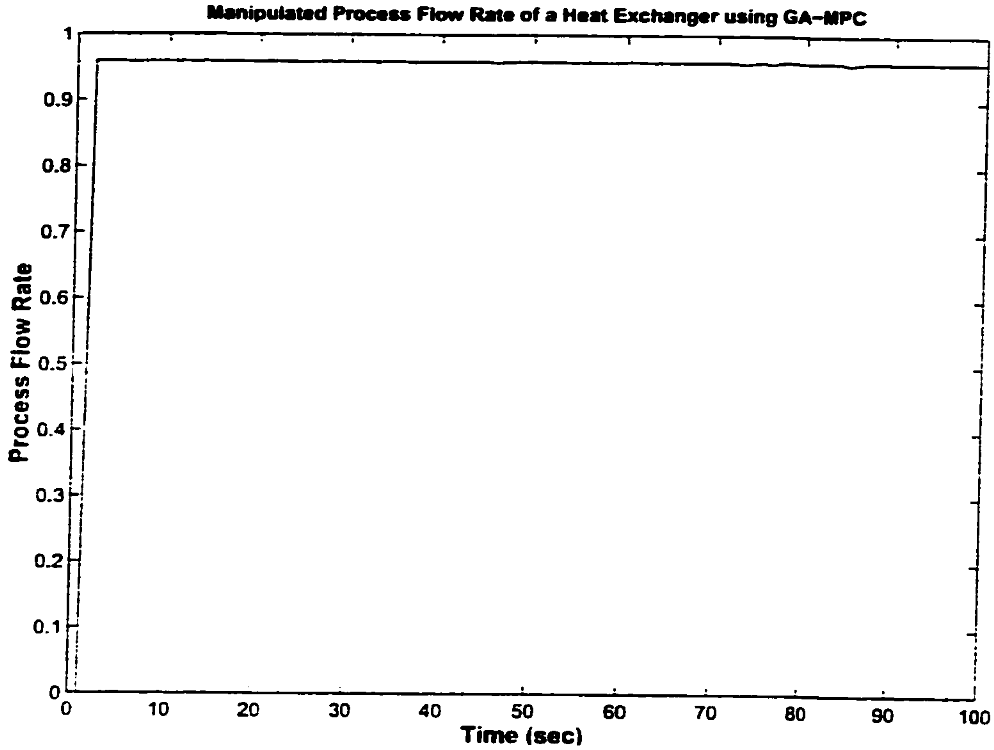

Figure 4.17: Optimal Inputs Generated by the controller for the Heat Exchanger.

4.2.3 GA-MPC of Linear MIMO Models

In this section, simulation results for the case of linear multi-input multi-output system is presented. The model taken is a 3-input 2-output linear model which is shown below.

$$\begin{bmatrix} 1 - 1.5q^{-1} + 0.7q^{-2} & 0.4 \\ -0.2q^{-1} + 0.01q^{-2} & 1 - 0.7q^{-2} \end{bmatrix} \begin{bmatrix} y_1(q^{-1}) \\ y_2(q^{-1}) \end{bmatrix} =$$

$$\begin{bmatrix} 0.2q^{-4} + 0.3q^{-5} & 0.4 - 0.1q^{-1} + 0.1q^{-2} & 0 \\ 1 & 2q^{-4} & 3q^{-1} + 4q^{-2} \end{bmatrix} \begin{bmatrix} u_1(q^{-1}) \\ u_2(q^{-1}) \end{bmatrix} \quad (4.9)$$

The parameters used in this simulation are given in Table 4.4.

	Population size	Number of Generations	Mutation Probability	Crossover Probability	Prediction Horizon(H_p)	Control Horizon(H_c)
Tracking	100	300	0.005	0.7	10	2

Table 4.4: Parameters used in the simulation of a Linear MIMO System.

The weighting matrices Q, R and S in the objective function are taken as

$$Q = I_2, \qquad R = 0.00002 I_3, \qquad S = 0.07 I_3$$

where

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and the input constraints are

$$-1 \leq u_1 \leq 3$$

$$-1 \leq u_2 \leq 5$$

$$-1 \leq u_3 \leq 1$$

For this system, constraints on the rate of change of inputs are also considered which are taken for all inputs as

$$-0.3 \leq \Delta u_i \leq 0.3, \qquad i = 1, 2, 3$$

The two outputs of the process are shown in Figures 4.18 and 4.19 and the three optimal inputs generated by the controller are shown in Figures 4.20, 4.21 and 4.22. The outputs shows good tracking performance while the inputs are converging to their optimal values. There are some oscillations in the beginning, however, the algorithm did well in suppressing the oscillations. There is also a large overshoot which can minimized by proper selection of the rate of change of input constraint or the tuning parameters in the objective function.

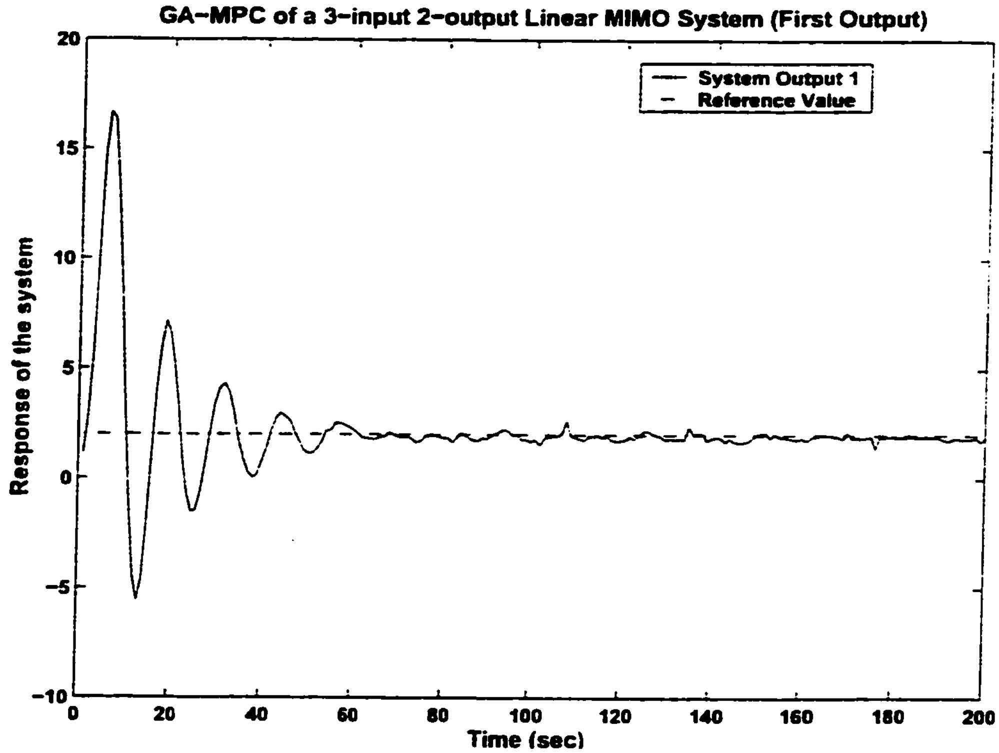

Figure 4.18: Application of the proposed algorithm to a 3-input 2-output Linear Model (First Output).

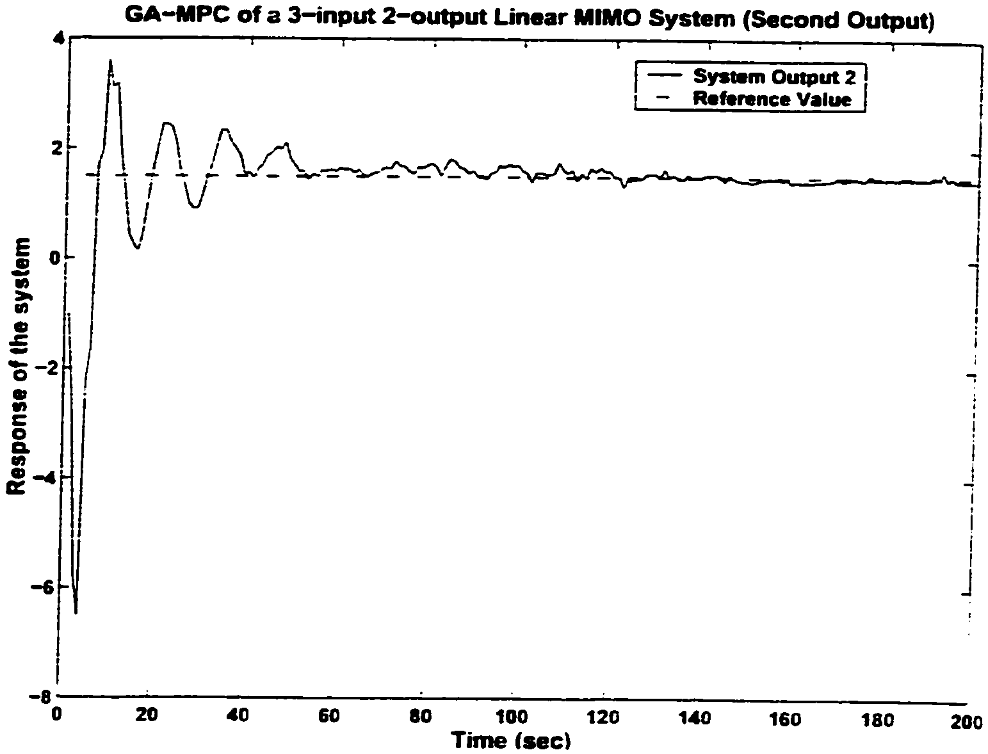

Figure 4.19: Application of the proposed algorithm to a 3-input 2-output Linear Model (Second Output).

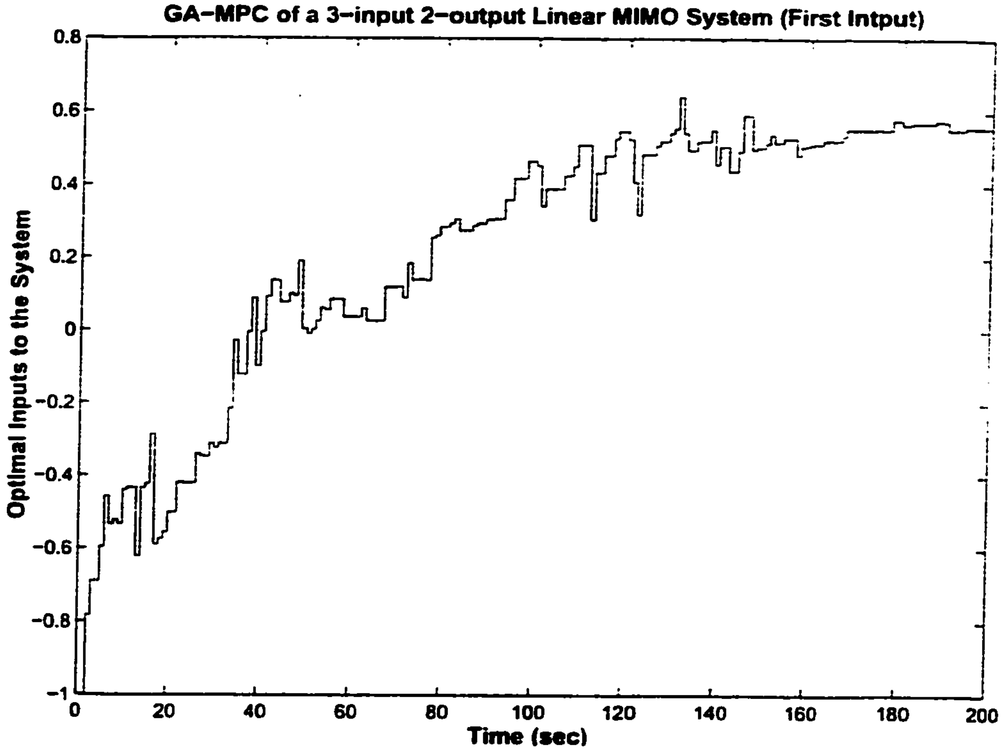

Figure 4.20: Optimal Inputs generated by the controller for a 3-input 2-output Linear Process (First Input).

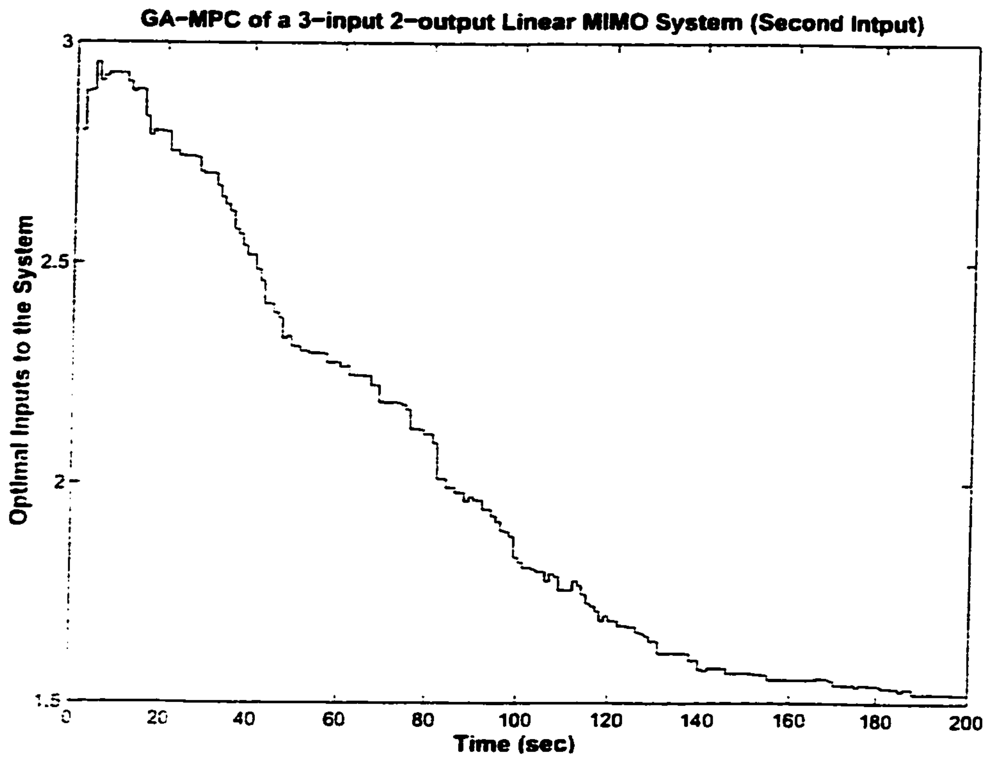

Figure 4.21: Optimal Inputs generated by the controller for a 3-input 2-output Linear Process (Second Input).

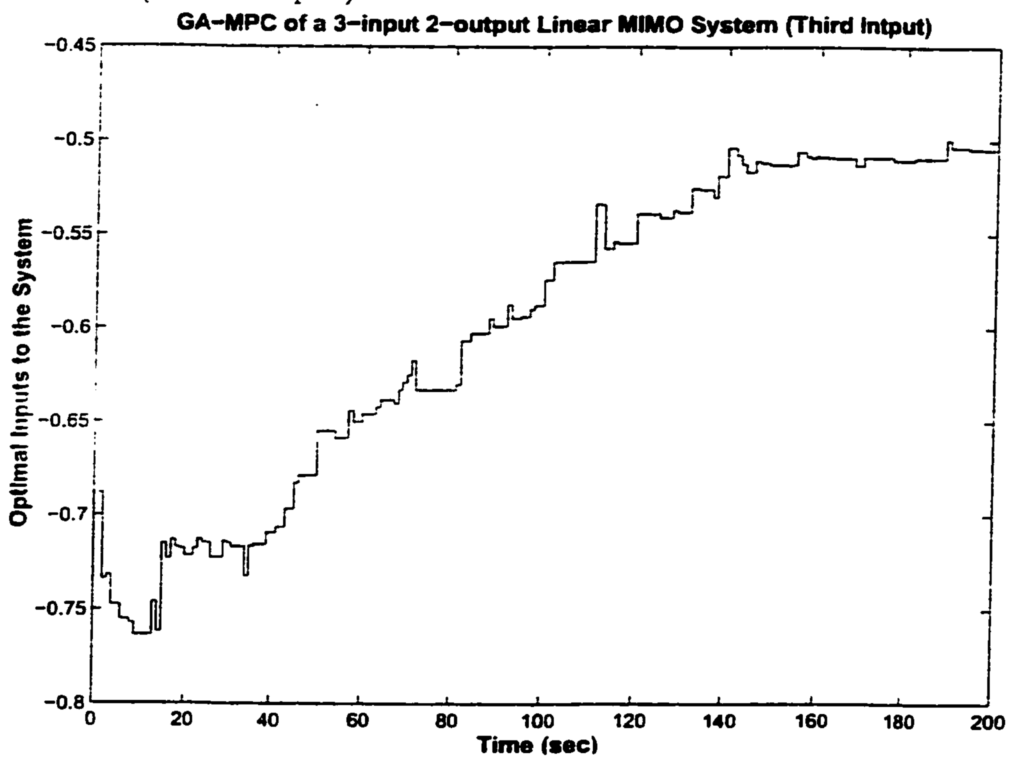

Figure 4.22: Optimal Inputs generated by the controller for a 3-input 2-output Linear Process (Third Input).

4.2.4 GA-MPC of Nonlinear MIMO Processes

First Example

For the simulation of nonlinear MIMO process. a two input. two output hammerstein model has been chosen. The model was obtained from [39] and is given by

$$A(q^{-1})Y(t) = B(q^{-1})X(t) \tag{4.10}$$

with

$$A(q^{-1}) = I + A_1 q^{-1} \tag{4.11}$$

$$B(q^{-1}) = (\xi(q))^{-1}(B_0 + B_1 q^{-1}) \tag{4.12}$$

The linear dynamic part is

$$A_1 = \begin{bmatrix} 0.10 & 0 \\ 0 & 0.09 \end{bmatrix}, B_1 = \begin{bmatrix} 0.199 & 0.612 \\ 0 & 0.798 \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 0.290 & 0.695 \\ 0 & 0.702 \end{bmatrix}$$

and

$$\xi(q) = \begin{bmatrix} q & 0 \\ 0 & q \end{bmatrix}$$

The static nonlinearities are

$$x_1(t) = 0.1 + u_1(t) + 0.7u_1(t)^3 \tag{4.13}$$

$$x_2(t) = -0.2 + 0.805u_2(t) + 0.58u_2(t)^3 \tag{4.14}$$

Both tracking and servo performances are shown for this system. The set points for the tracking case are chosen as 0.5 and 0.8 for the two outputs. For the servo performance, the set points are chosen as [0.5, 1.2] and [1.2, 0.5]. The input constraints are

$$-1 \leq u_1 \leq 1$$
$$-1 \leq u_2 \leq 1$$

The parameters used in the simulation are provided in Table 4.5. and the weighting matrices are taken as

$$Q = I_2, \qquad R = O_2, \qquad S = O_2$$

where

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad O_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

	Population size	Number of Generations	Mutation Probability	Crossover Probability	Prediction Horizon(H_p)	Control Horizon(H_c)
Tracking	100	50	0.005	0.7	3	3
Servo	100	50	0.005	0.7	3	3

Table 4.5: Parameters used in the simulation of a MIMO Hammerstein Model.

Figures 4.23 and 4.24 show the tracking performance of the algorithm and Figure 4.27 and 4.28 show the servo performance. The algorithm did not find any problem in tracking the set point. The inputs for the two cases are also shown.

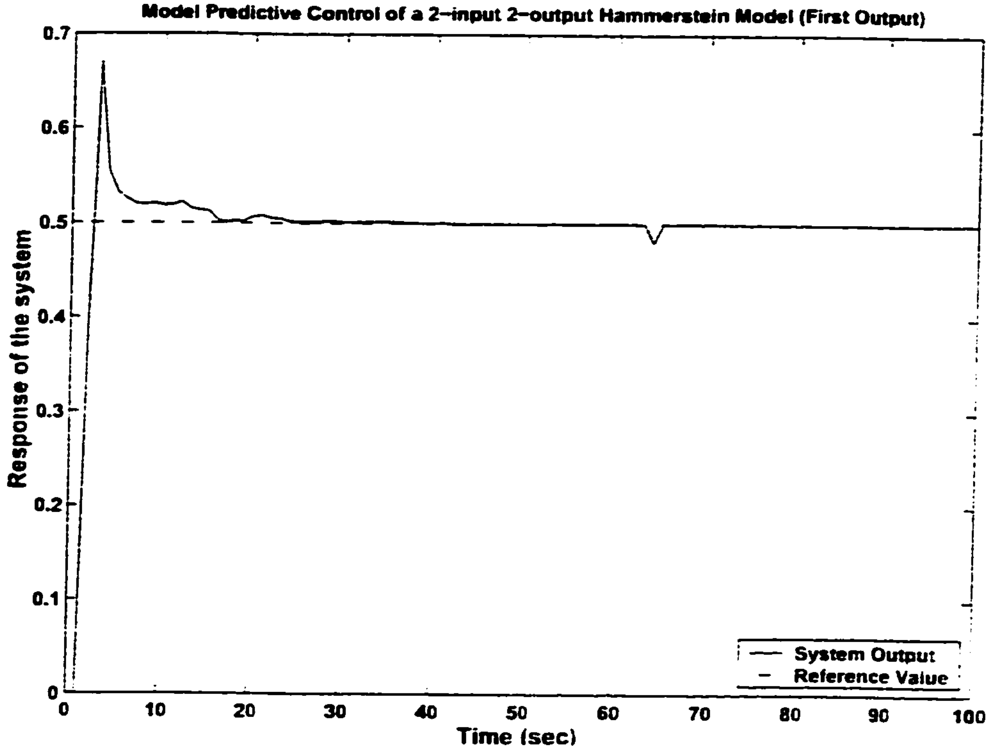

Figure 4.23: Application of the Proposed Algorithm to Nonlinear 2-input 2-output Hammerstein Model (First Output). Tracking Case.

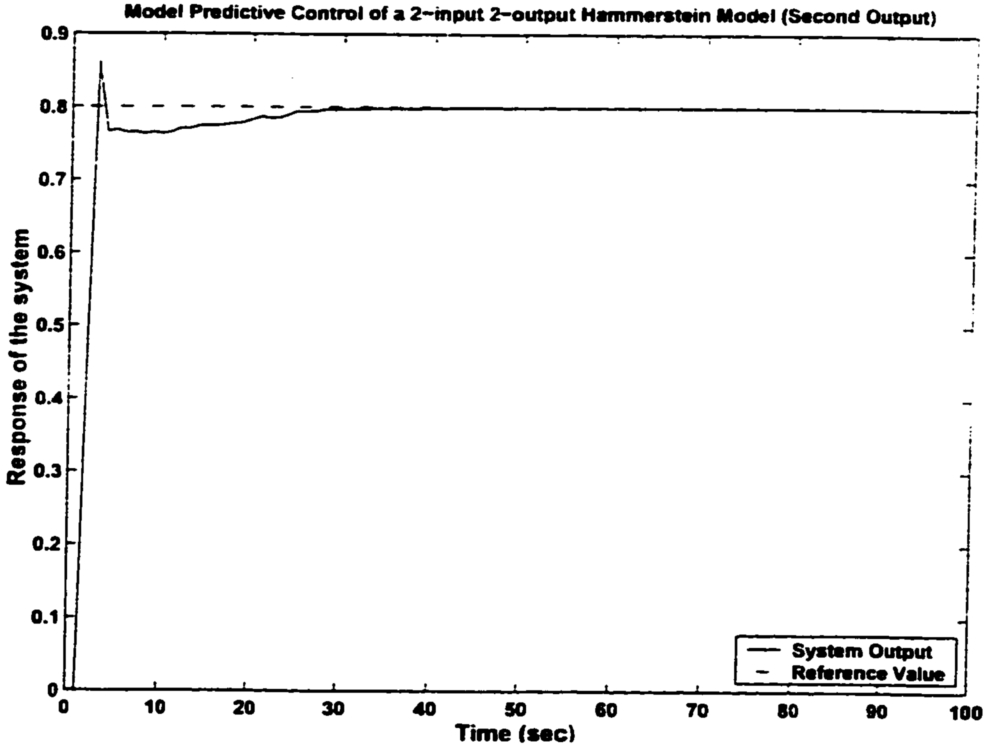

Figure 4.24: Application of the Proposed Algorithm to Nonlinear 2 input 2 output Hammerstein Model (Second Output). Tracking Case.

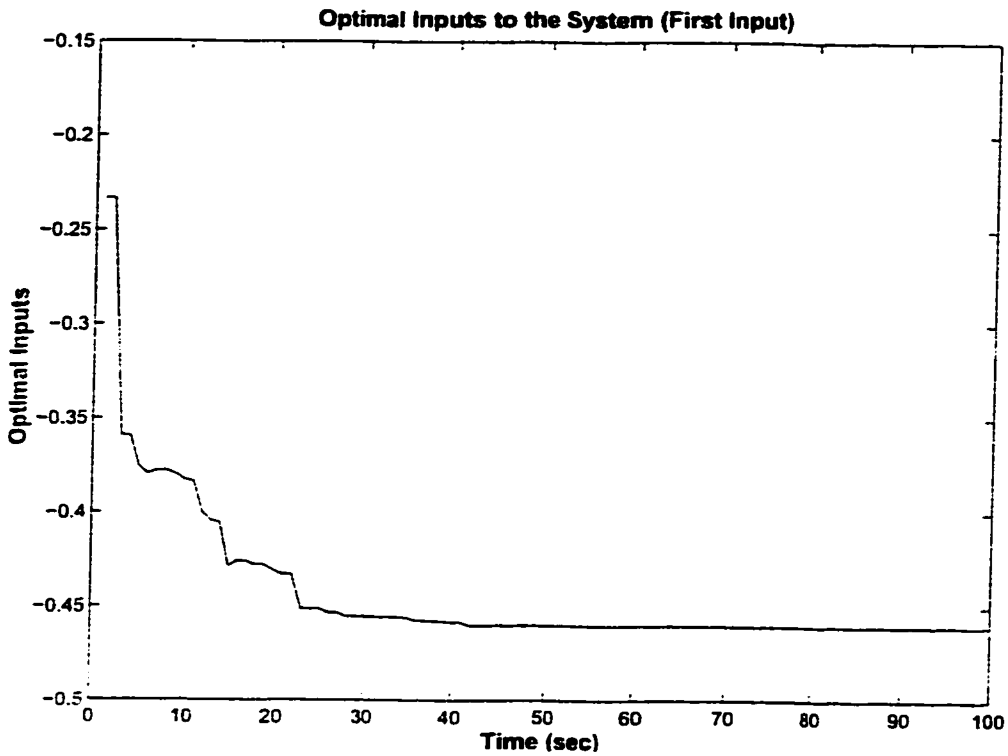

Figure 4.25: Optimal Inputs generated by the controller for a Nonlinear 2-input 2-output Hammerstein Model (First Input). Tracking Case.

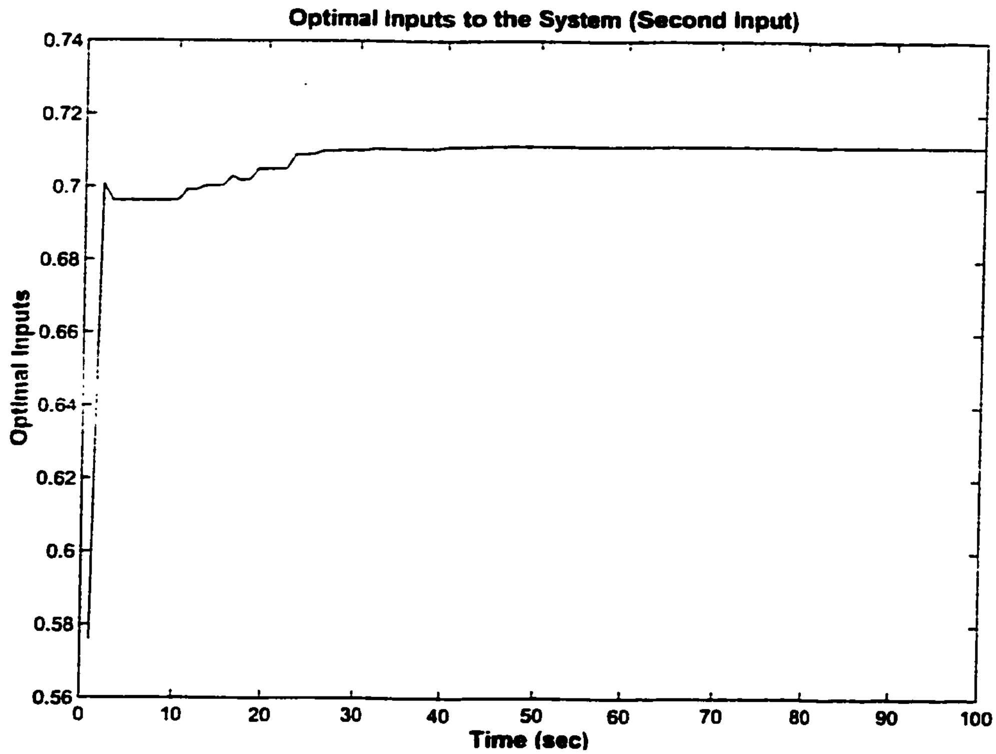

Figure 4.26: Optimal Inputs generated by the controller for a Nonlinear 2-input 2-output Hammerstein Model (Second Input). Tracking Case.

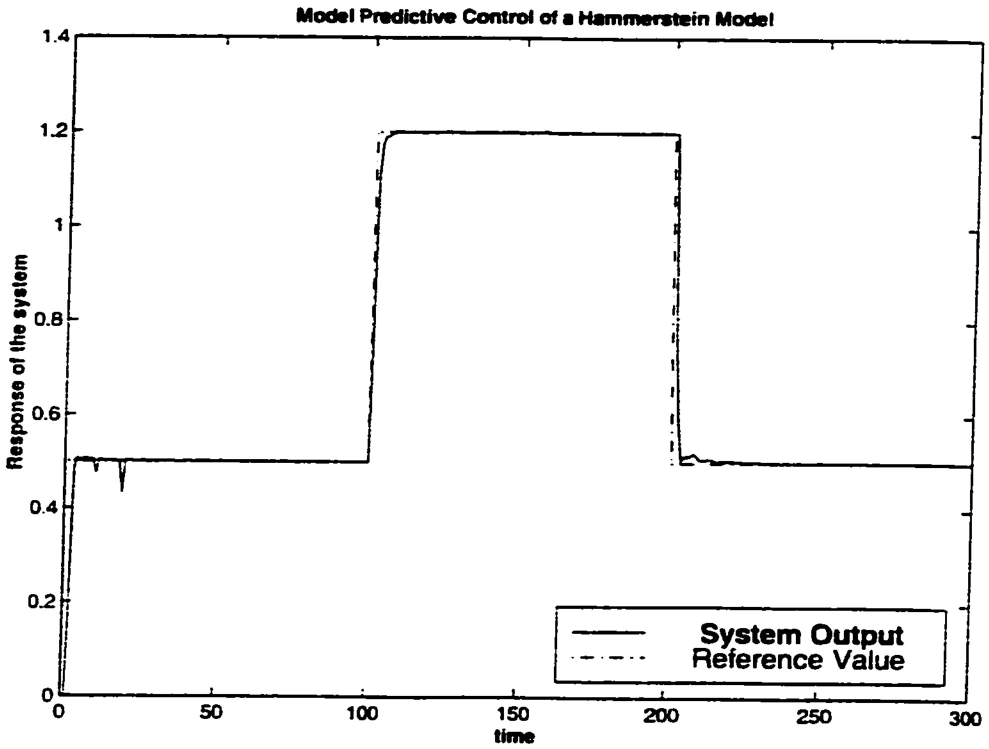

Figure 4.27: Application of the Proposed Algorithm to Nonlinear 2 input 2 output Hammerstein Model (First Output) Servo Case.

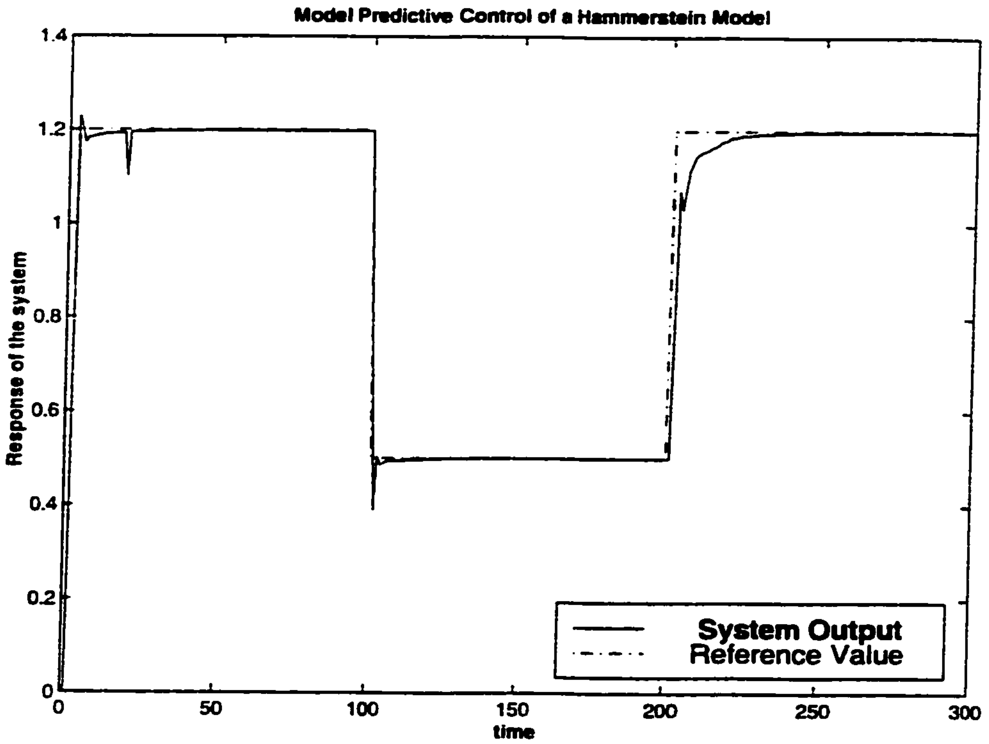

Figure 4.28: Application of the Proposed Algorithm to Nonlinear 2 input 2 output Hammerstein Model (Second Output) Servo Case.

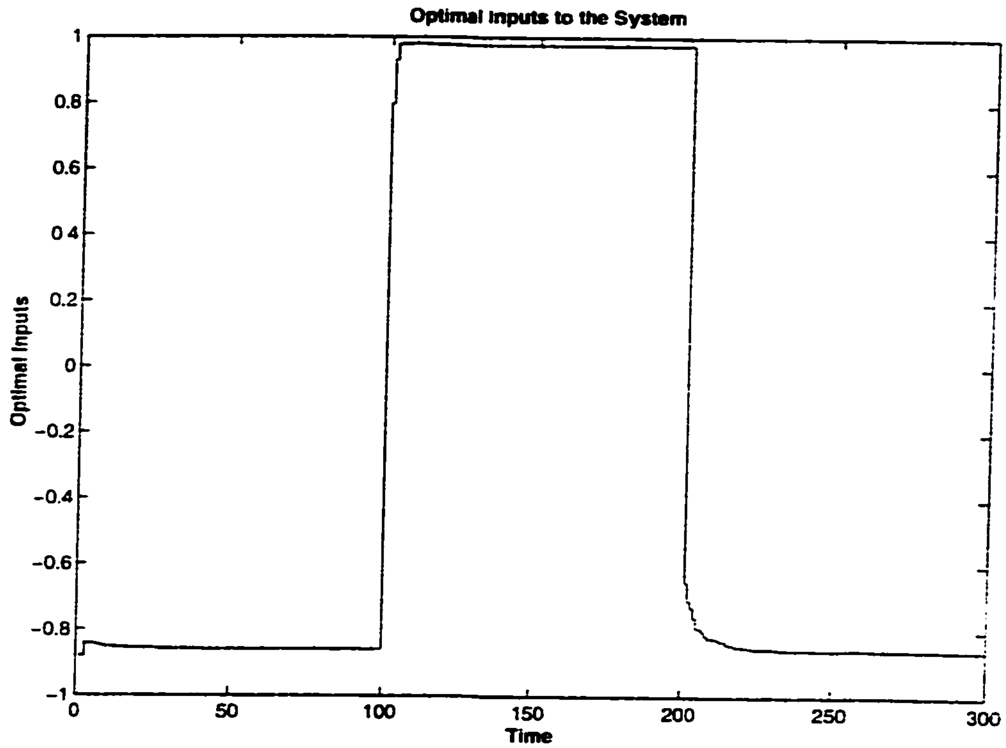

Figure 4.29: Optimal Inputs generated by the controller for a Nonlinear 2-input 2-output Hammerstein Model (First Input) Servo Case.

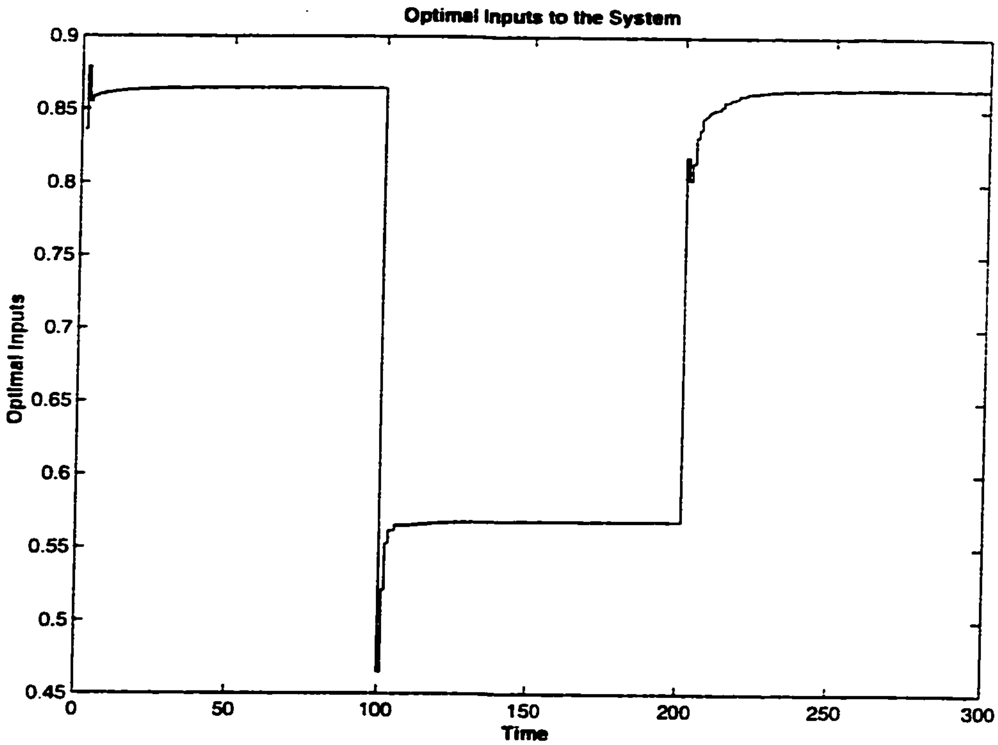

Figure 4.30: Optimal Inputs generated by the controller for a Nonlinear 2-input 2-output Hammerstein Model (Second Input) Servo Case.

Second Example- Case Study of a Binary Distillation Column

This section examines the control of an ideal binary distillation column as an example of a MIMO nonlinear system. The reflux flow and the vapor boilup are the manipulated variables used to control the top composition and the bottom composition. The steady state properties of the distillation column are 0.98 and 0.02 for the top composition and bottom composition and 128.01 and 178.01 for the reflux flow and vapor boilup respectively. The model has been obtained from [21], which describes the distillation column as a 2-input 2-output hammerstein model. The nonlinearities in this case are commonly parameterized as contrast to the previous case in which the nonlinearities were seperately parameterized. The linear dynamics is shown in Equation 4.15.

$$y_1 = \frac{-0.0157q^{-1}}{1-0.9522q^{-1}}x_1 + \frac{-0.0047}{1+0.02754q^{-1}}x_2 \quad (4.15)$$

$$y_2 = \frac{-0.0201q^{-1}}{1-0.9060q^{-1}}x_1 + \frac{-0.0302}{1-0.9991q^{-1}}x_2 \quad (4.16)$$

while the commonly parameterized nonlinearities are given by

$$x_1 = u_1 - 0.0236u_1u_2 \quad (4.17)$$

$$x_2 = u_2 - 0.1823u_1u_2 \quad (4.18)$$

The parameters used in the simulation for the distillation column example are presented in Table 4.6. The weighting matrices used in the objective function are taken

	Population size	Number of Generations	Mutation Probability	Crossover Probability	Prediction Horizon(H_p)	Control Horizon(H_c)
Tracking	200	10	0.002	1	50	1

Table 4.6: Parameters used in the simulation of a Distillation Column.

as

$$Q = I_2, \qquad R = O_2, \qquad S = O_2$$

where

$$O_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Constraints imposed on the inputs are

$$-0.4 \leq u_1 \leq 0.4$$

$$-0.4 \leq u_2 \leq 0.4$$

Figures 4.31 and 4.32 show the dynamics of the bottom product and the top composition using the *Hammerstein model.*

The overshoot for *bottom composition* is quite high but this can be minimized by selecting proper tuning parameters in the objective function and also by limiting the rate of change of inputs by imposing constraints on it.

The *top composition* tracks the set point well and there is no overshoot. Looking at the input plots, the reflux becomes saturated for a short time when its lower constraint becomes active. However, the controller never allows the input to violate its constraints.

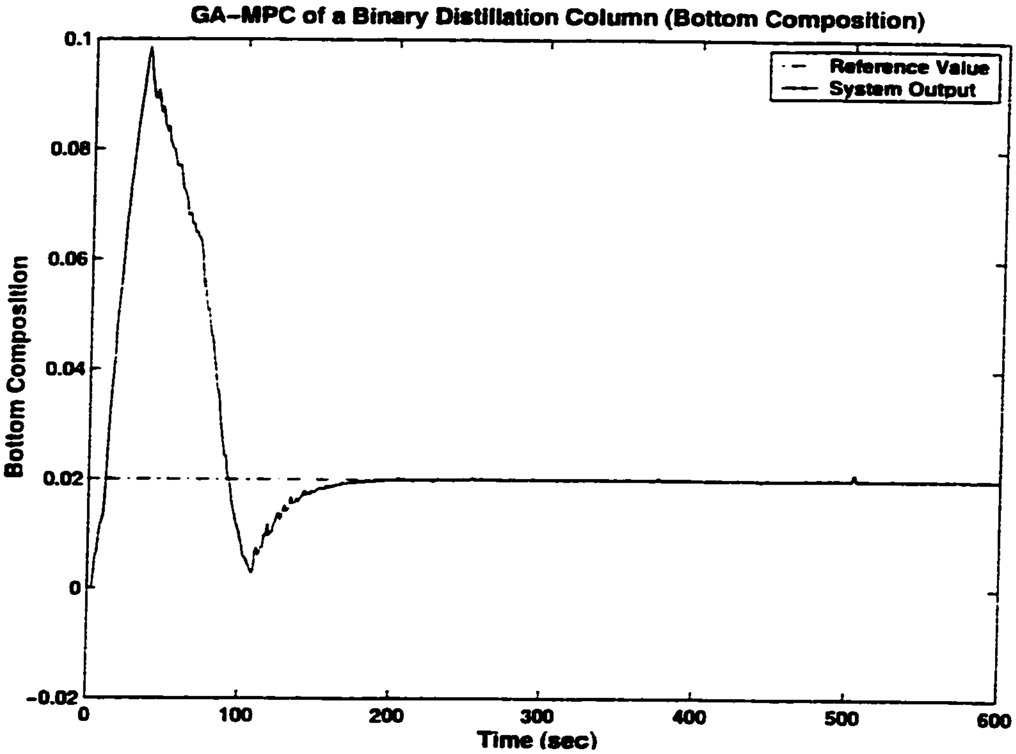

Figure 4.31: Application of the proposed algorithm to Nonlinear 2-input 2-output Distillation Column (Bottom Composition).

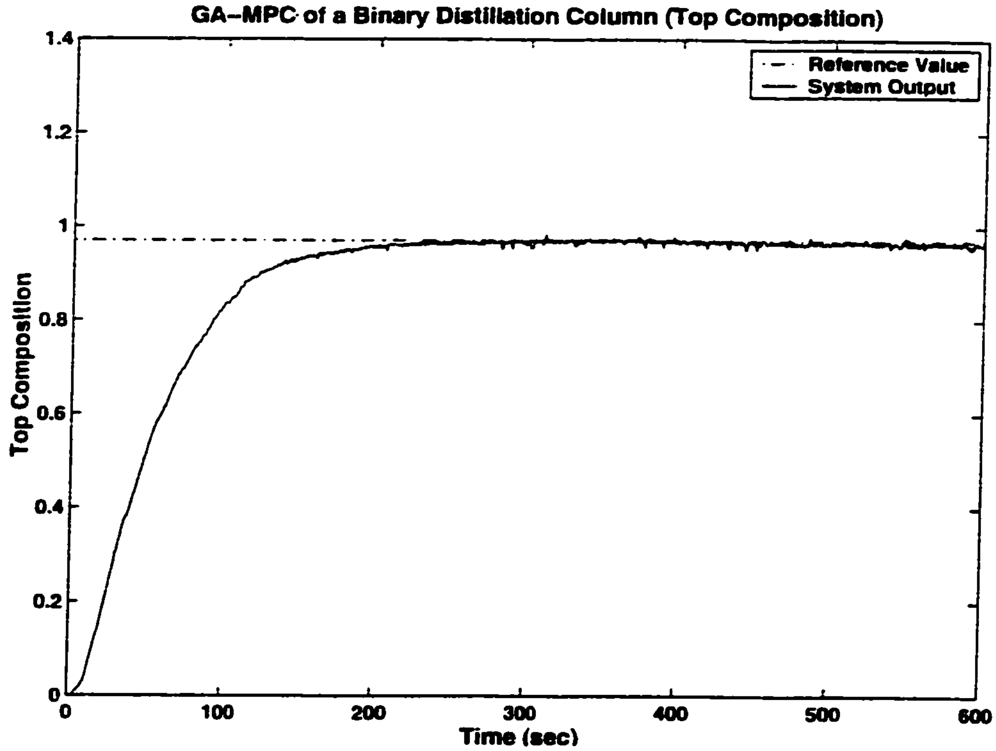

Figure 4.32: Application of the proposed algorithm to Nonlinear 2-input 2-output Distillation Column (Top Composition).

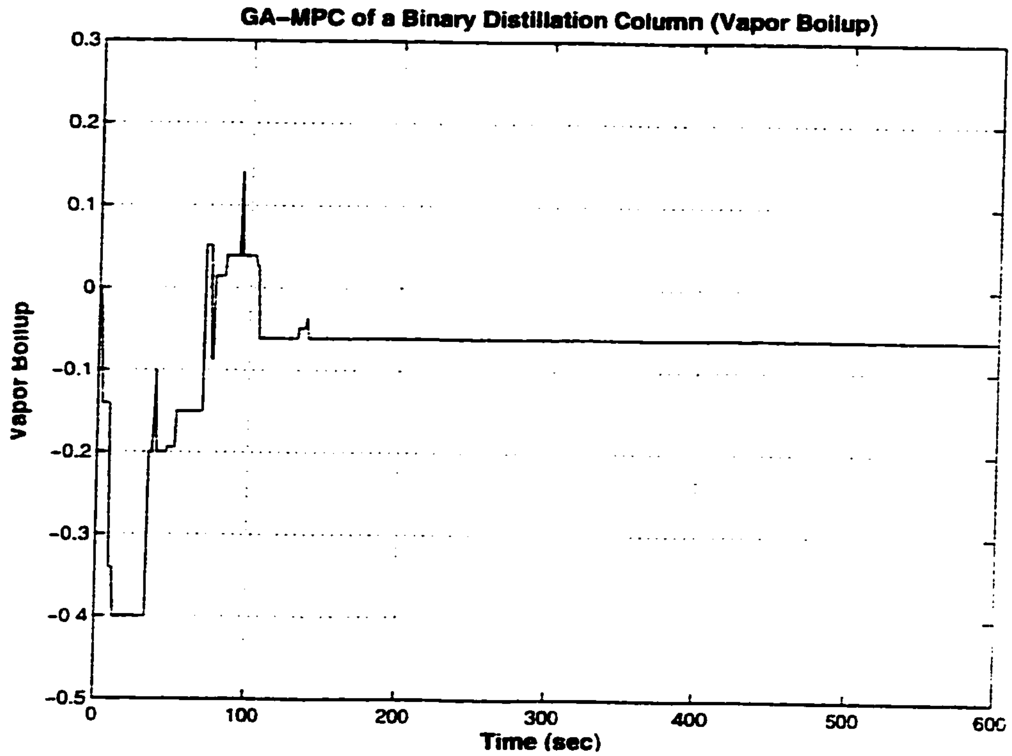

Figure 4.33: Optimal Inputs generated by the controller for a Nonlinear 2-input 2-output Distillation Column (Vapor Boil-up).

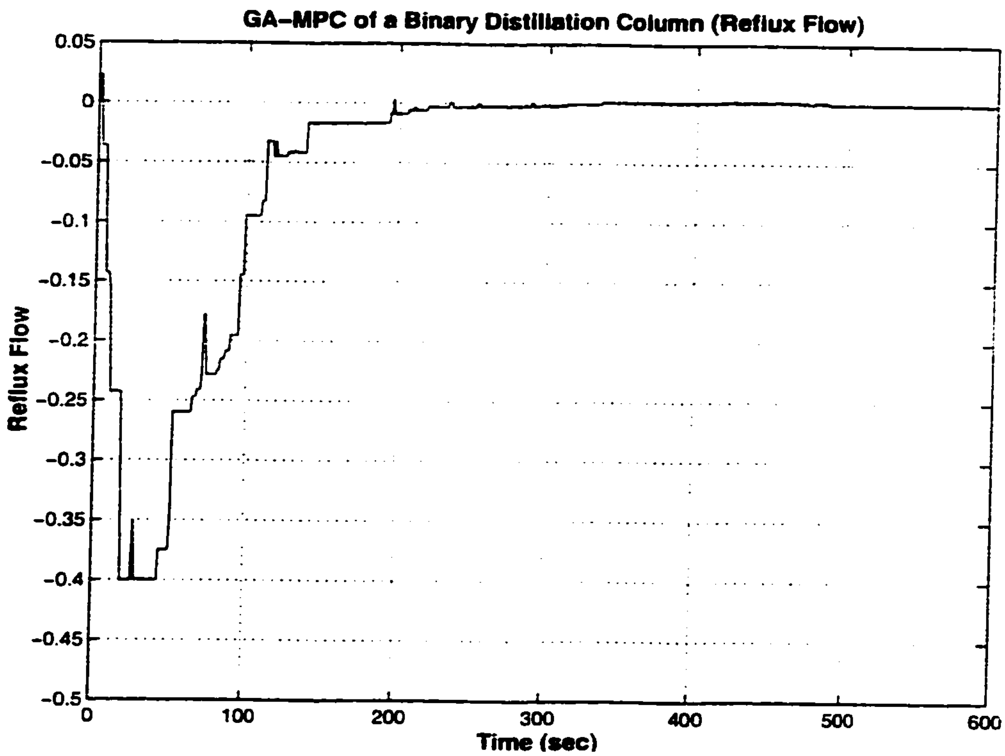

Figure 4.34: Optimal Inputs generated by the controller for a Nonlinear 2-input 2-output Distillation Column (Reflux Flow).

Third Example- Case Study of a Power System

The system simulated in this section is a single-machine infinite-bus power system installed with a STATCOM (Static Synchronous Compensator) and has been borrowed from [40]. It is a continuous time system and is discretized at a sampling rate of 1 *sec*. This example is a Hammerstein model with two inputs and two outputs. The linear system is of fifth order and is given in the state space form in Equations 4.19 and 4.20.

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} - \mathbf{B}\mathbf{x} \tag{4.19}$$

$$\mathbf{y} = \mathbf{C}\mathbf{z} + \mathbf{D}\mathbf{x} \tag{4.20}$$

where.

$$\mathbf{z} = \begin{bmatrix} \delta & \omega & E_q' & E_{fd} & v_{dc} \end{bmatrix}^T, \mathbf{x} = \begin{bmatrix} c \\ \psi \end{bmatrix}, \mathbf{y} = \begin{bmatrix} E_q' \\ v_{dc} \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 377 & 0 & 0 & 0 \\ -0.0576 & -0.6667 & -0.1717 & 0 & -0.0128 \\ -0.0458 & 0 & -0.3175 & 0.1587 & 0.0205 \\ 51.4152 & 0 & -515.1560 & -100 & -77.2599 \\ -0.5292 & 0 & 0.1503 & 0 & -0.0058 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ -0.2 & 0.1 \\ 0.3 & 0.1 \\ -1271.3 & -29.1 \\ -0.1 & -0.7 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$\delta \longrightarrow$ Rotor angle

$\omega \longrightarrow$ Angular frequency

$E'_q \longrightarrow$ Voltage behind transient reactance

$E_{fd} \longrightarrow$ Internal Excitation Voltage

$v_{dc} \longrightarrow$ DC Voltage

$c = mk$. where k is the ratio between AC and DC voltage and m is the modulation ratio defined by the PWM

$\psi \longrightarrow$ phase defined by the PWM

The nonlinear part is a saturation nonlinearity given by Equation 4.21.

$$x(k) = \frac{u(k)}{\sqrt{0.10 + 0.90u^2(k)}} \tag{4.21}$$

The goal in this simulation is to minimize the oscillations of the power plant when a step disturbance occur at one of the inputs at a certain time. The outputs of the system under consideration are the rotor angle and DC Voltage. First the system is simulated without any rate constraints. The constraints on the inputs are

$$-1 \leq c \leq 1$$

$$-1 \leq \psi \leq 1$$

The weighting matrices Q. R. and S in the objective function are taken as

$$Q = I_2, \qquad R = 0.001I_2, \qquad S = 0.001I_2$$

where

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Other parameters used in the simulation are provided in Table 4.7.

Population size	Number of Generations	Mutation Probability	Crossover Probability	Prediction Horizon(H_p)	Control Horizon(H_c)
100	100	0.008	0.8	100	1

Table 4.7: Parameters used in the simulation of a Power system.

It is assumed that the system is initially at steady state and a step disturbance has occured at $k = 30$ on c for 2 sampling times as shown in Figure 4.37. The optimal inputs generated by the controller are shown in Figures 4.37 and 4.38, for which the outputs of the system obtained are shown in Figures 4.35 and 4.36. Clearly, both outputs settle down after some oscillations. The magnitude of the oscillations is quite large which can be minimized by incorporating the rate constraints at the inputs. For this purpose the rate constraints are taken on the inputs as

$$-0.05 \leq \Delta c \leq 0.05$$

$$-0.05 \leq \Delta\psi \leq 0.05$$

The system is simulated again with the same parameters as before and the outputs are shown in Figures 4.39 and 4.40. Again the outputs settle down after some oscillations but this time the magnitude of the oscillations subsides considerably. The inputs generated by the controller are also shown to be in the constrained limits in Figures 4.41 and 4.42.

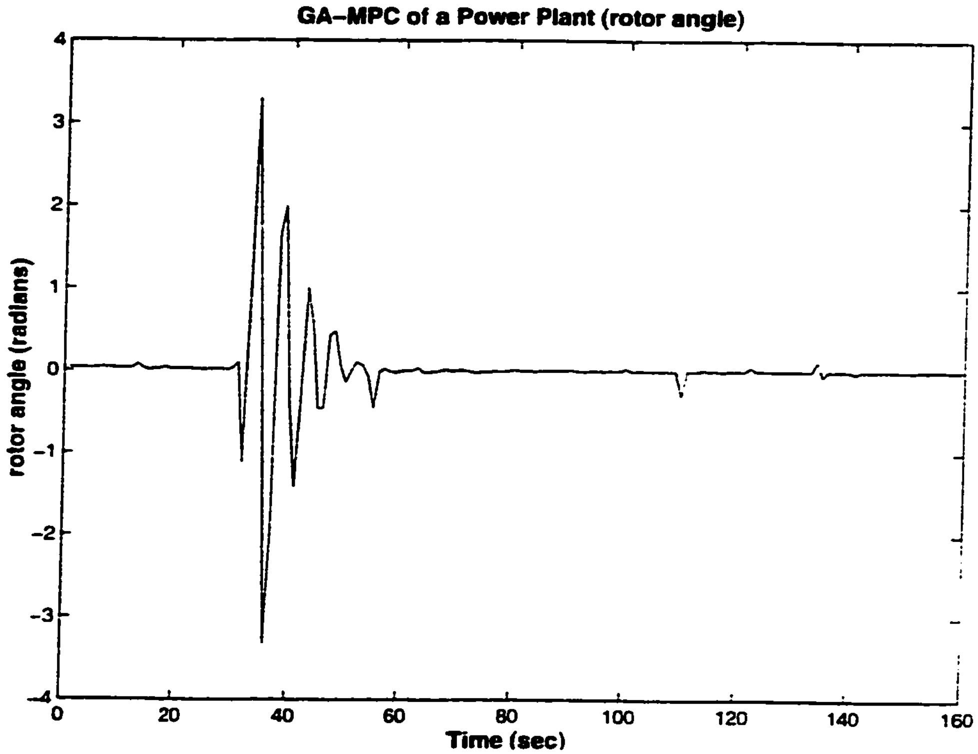

Figure 4.35: GA-MPC of a Power Plant with no Rate Constraints (Rotor Angle).

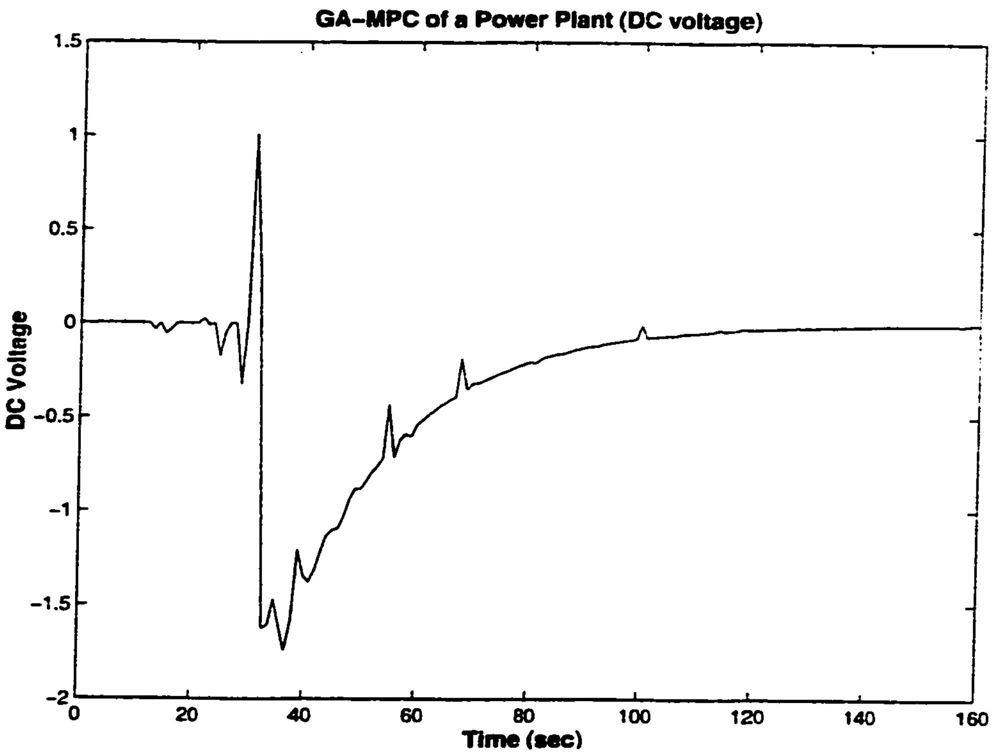

Figure 4.36: GA-MPC of a Power Plant with no Rate Constraints (DC Voltage).

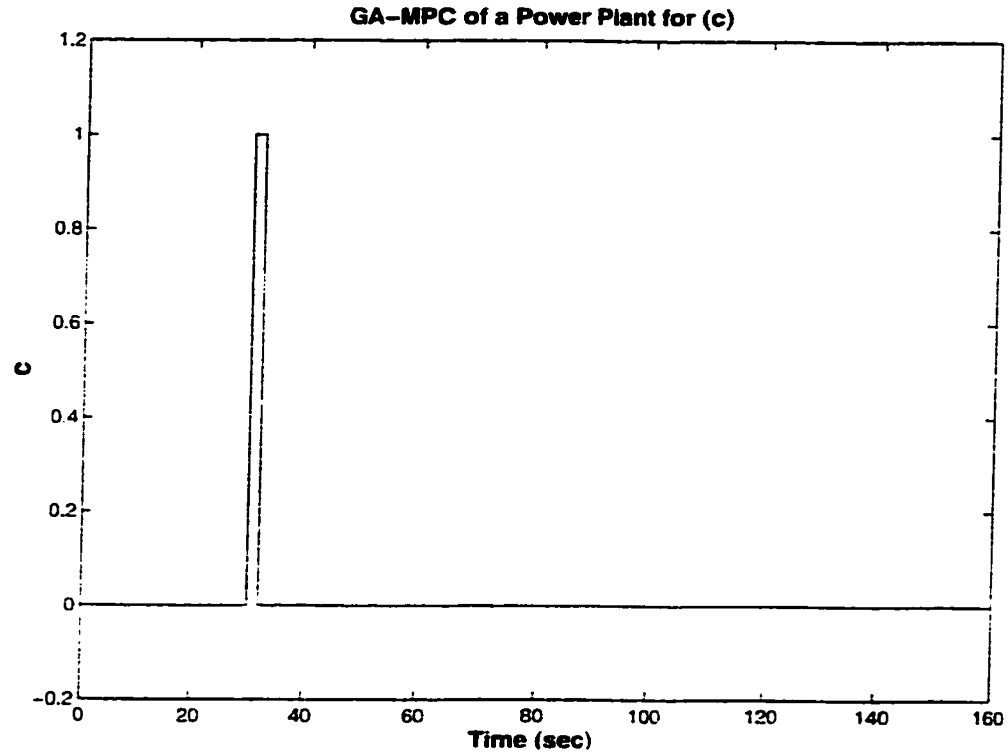

Figure 4.37: GA-MPC of a Power Plant with no Rate Constraints (c).

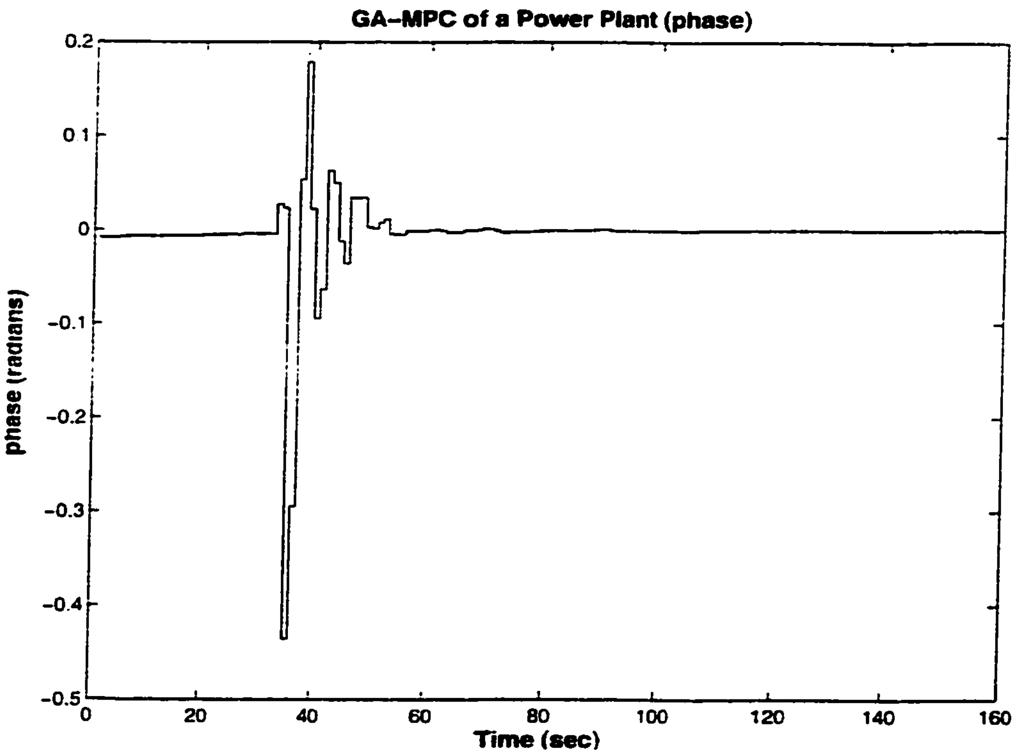

Figure 4.38: GA-MPC of a Power Plant with no Rate Constraints (Phase).

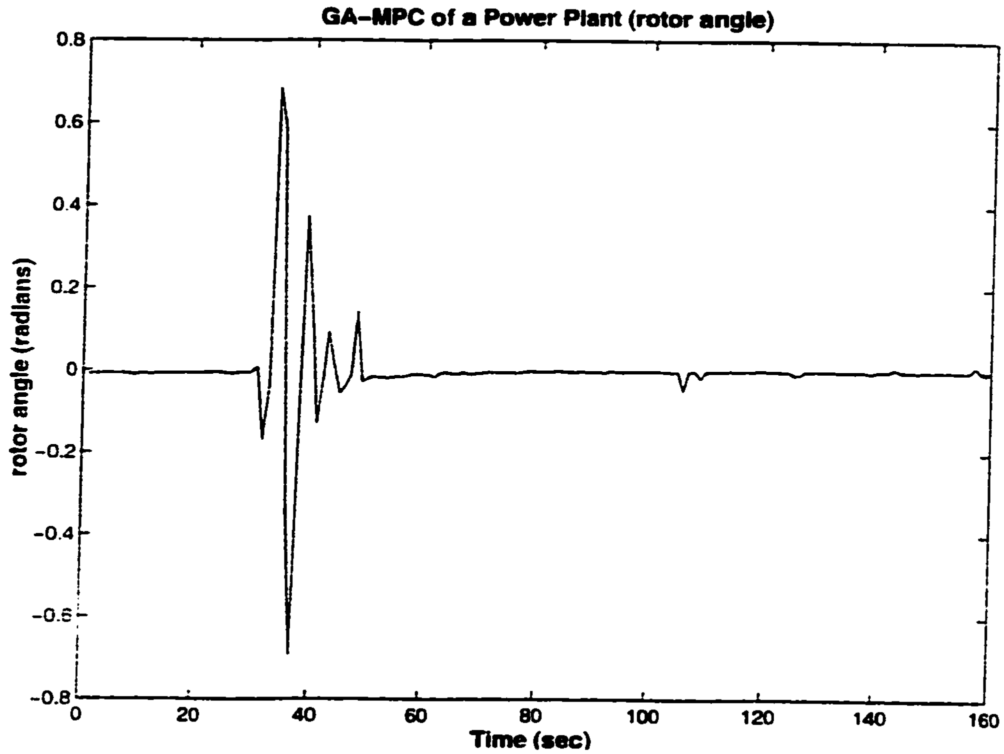

Figure 4.39: GA-MPC of a Power Plant with Rate Constraints (Rotor Angle).

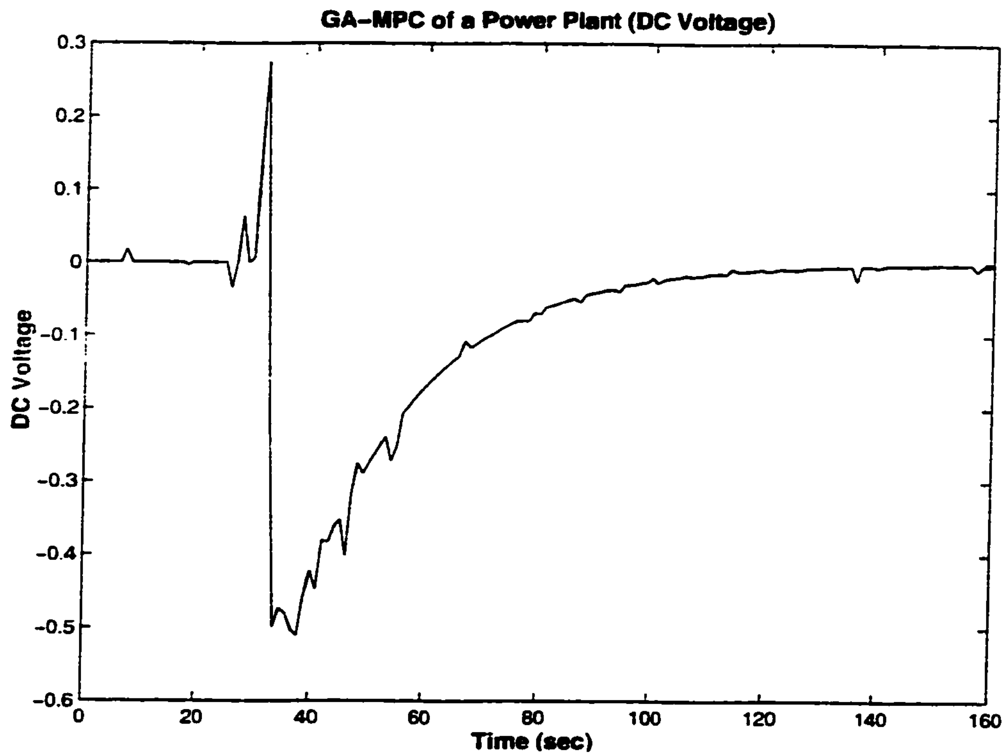

Figure 4.40: GA-MPC of a Power Plant with Rate Constraints (DC Voltage).

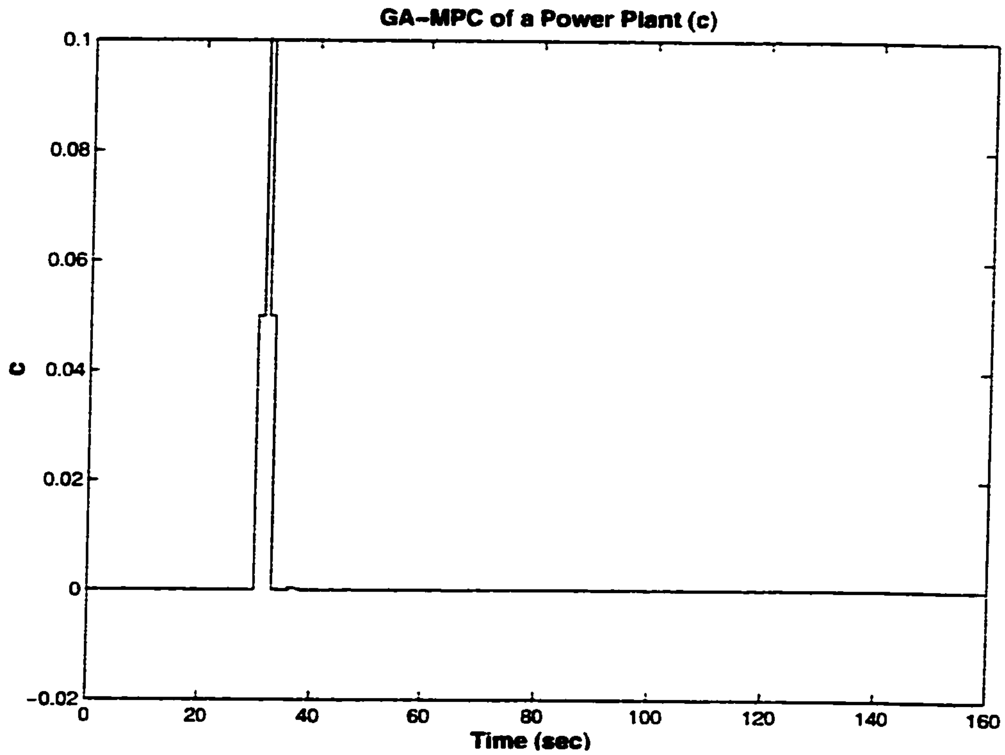

Figure 4.41: GA-MPC of a Power Plant with Rate Constraints (c).

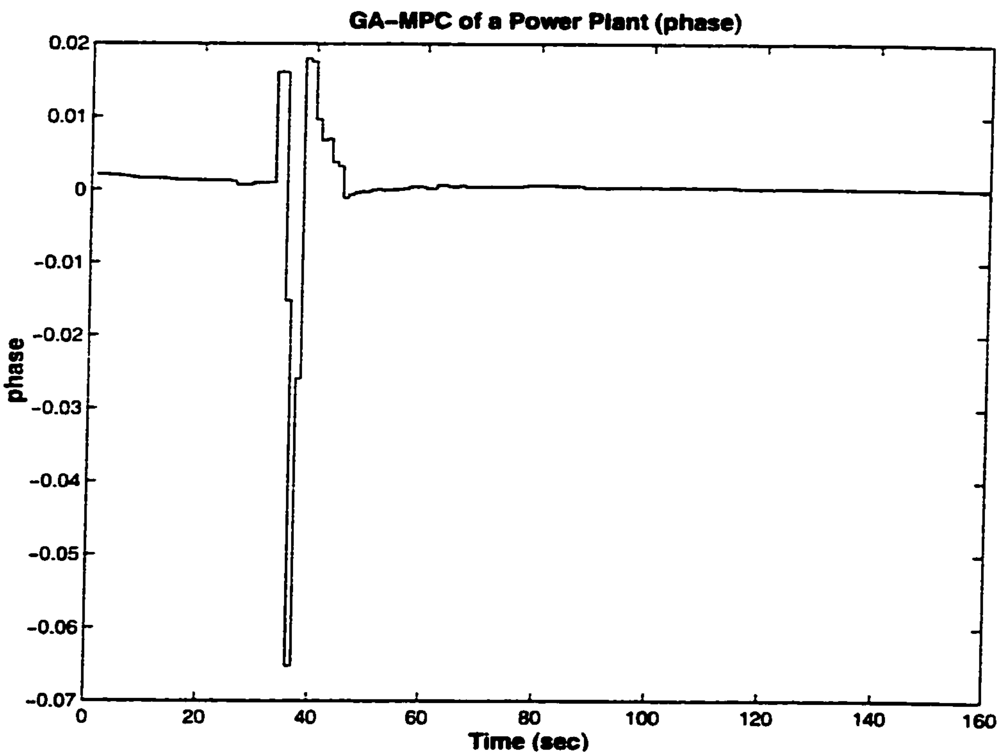

Figure 4.42: GA-MPC of a MIMO Power Plant with Rate Constraints (Phase).

4.2.5 GA-MPC Performance in Noisy Environment

In this section, the performance of the proposed algorithm in noisy environment is considered.

The block diagram for this case for a hammerstein model is shown in Figure 4.43. The example from Section 4.2.4 is taken . but now a noise term is augmented to it.

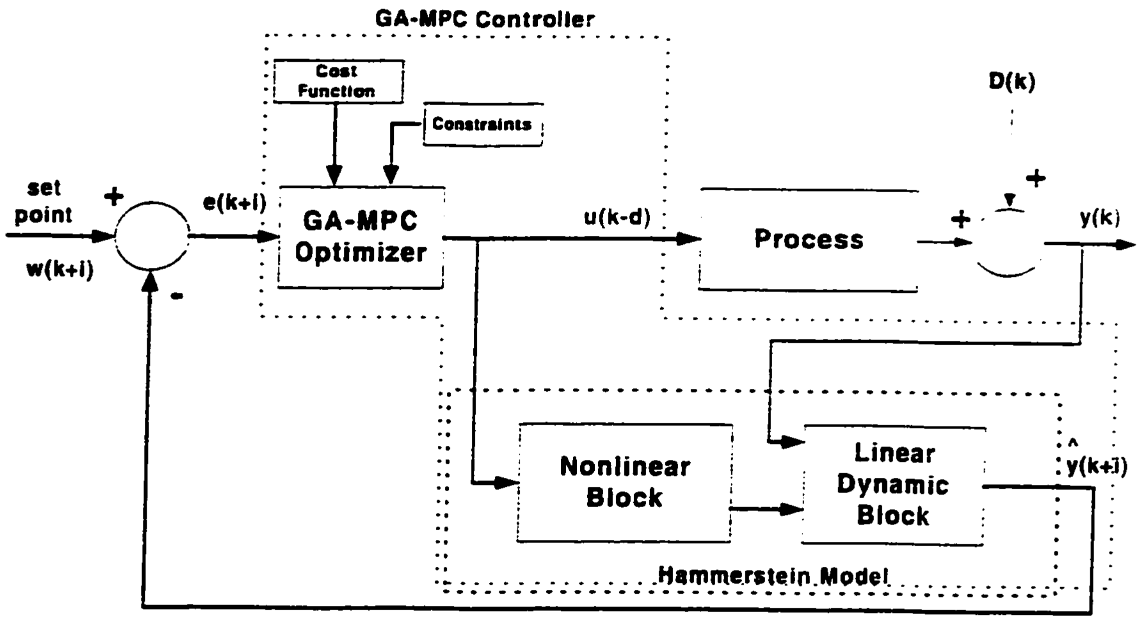

Figure 4.43: Proposed GA-based MBPC of a Hammerstein model with Noise.

The process model is given by

$$A(q^{-1})Y(k) = B(q^{-1})X(k) + C(q^{-1})D(k) \tag{4.22}$$

with

$$A(q^{-1}) = I + A_1 q^{-1} \quad (4.23)$$

$$B(q^{-1}) = (\xi(q))^{-1}(B_0 + B_1 q^{-1}) \quad (4.24)$$

$$C(q^{-1}) = I + C_1(q^{-1}) \quad \text{and} \quad (4.25)$$

$$D(q^{-1}) = I + D_1 q^{-1} \quad (4.26)$$

The linear dynamic part is

$$A_1 = \begin{bmatrix} 0.10 & 0 \\ 0 & 0.09 \end{bmatrix}, B_1 = \begin{bmatrix} 0.199 & 0.612 \\ 0 & 0.798 \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 0.290 & 0.695 \\ 0 & 0.702 \end{bmatrix}, C_1 = \begin{bmatrix} -1.86 & 1.099 \\ -1.92 & -1.533 \end{bmatrix}$$

and

$$\xi(q) = \begin{bmatrix} q & 0 \\ 0 & q \end{bmatrix}$$

The static nonlinearities are

$$x_1(t) = 0.1 + u_1(t) + 0.7u_1(t)^3 \quad (4.27)$$

$$x_2(t) = -0.2 + 0.805u_2(t) + 0.58u_2(t)^3 \quad (4.28)$$

$D(k) = \lambda e(k)$, with λ the gain matrix or the variance of the noise and $e(k)$ the white noise.

$$\lambda = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

	Population size	Number of Generations	Mutation Probability	Crossover Probability	Prediction Horizon(H_p)	Control Horizon(H_c)
Tracking	100	100	0.005	0.7	3	1

Table 4.8: Parameters used in the simulation of a Hammerstein Model with Noise.

The parameters used in this case are provided in Table 4.8 The input constraints are

$$-1 \leq u_1 \leq 1$$

$$-1 \leq u_2 \leq 1$$

The algorithm is simulated and the outputs are shown in Figures 4.44 and 4.45. The controller output or the process inputs are shown in Figures 4.46 and 4.47.

The controller still able to track the set point despite the presence of the noise. However. there are oscillations present about the set point. The inputs are also converging to their optimal values.

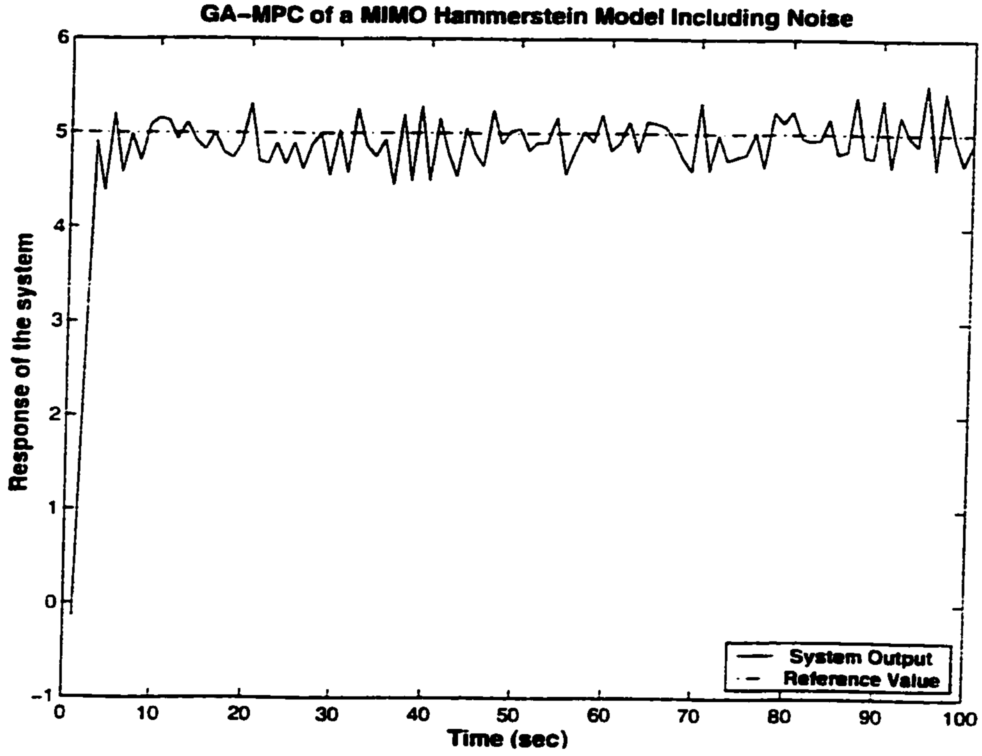

Figure 4.44: Application of the proposed algorithm to a Nonlinear 2-input 2-output Hammerstein Model considering the noise (First Output).

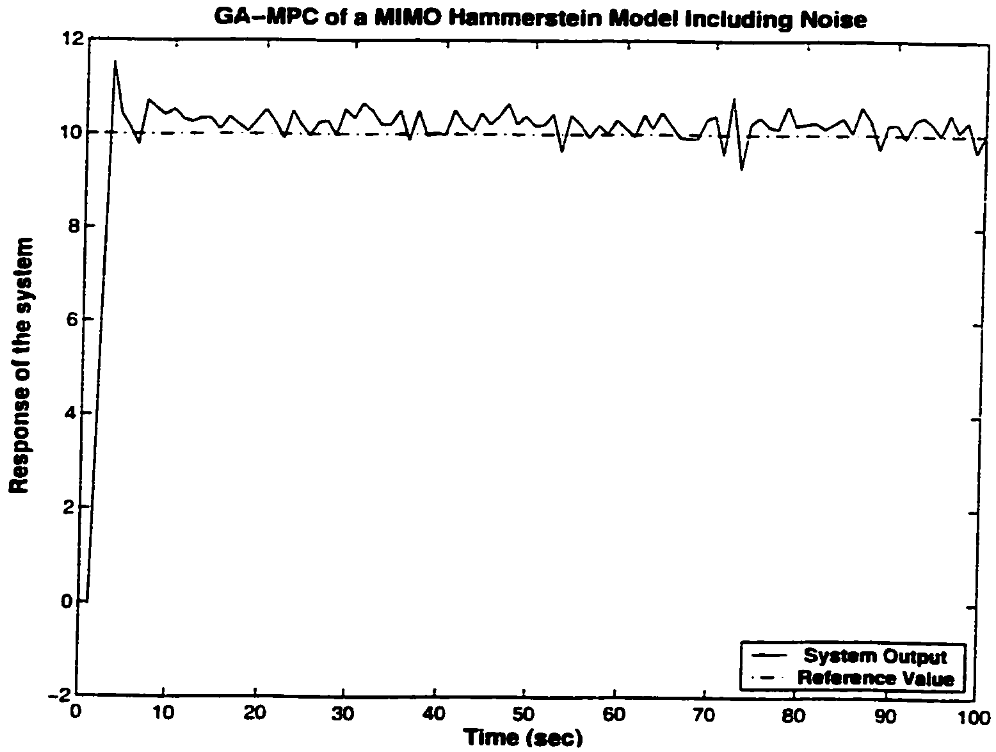

Figure 4.45: Application of the proposed algorithm to a Nonlinear 2-input 2-output Hammerstein Model considering the noise (Second Output).

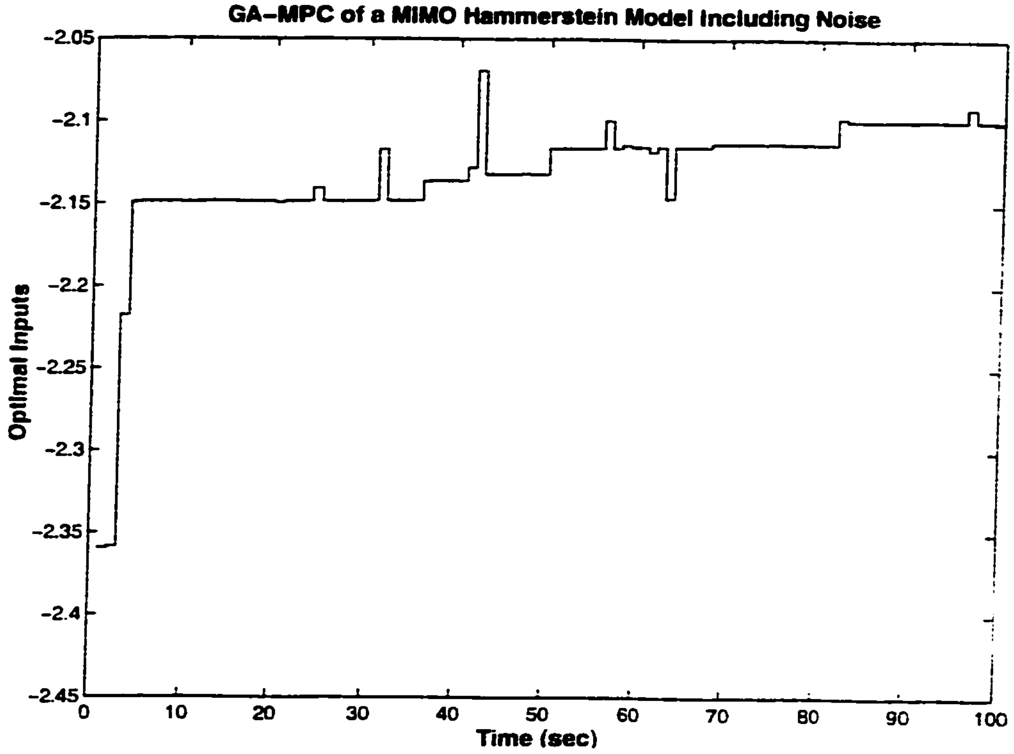

Figure 4.46: Optimal Inputs generated by the controller for a 2-input 2-output Hammerstein Model including noise (First Input).

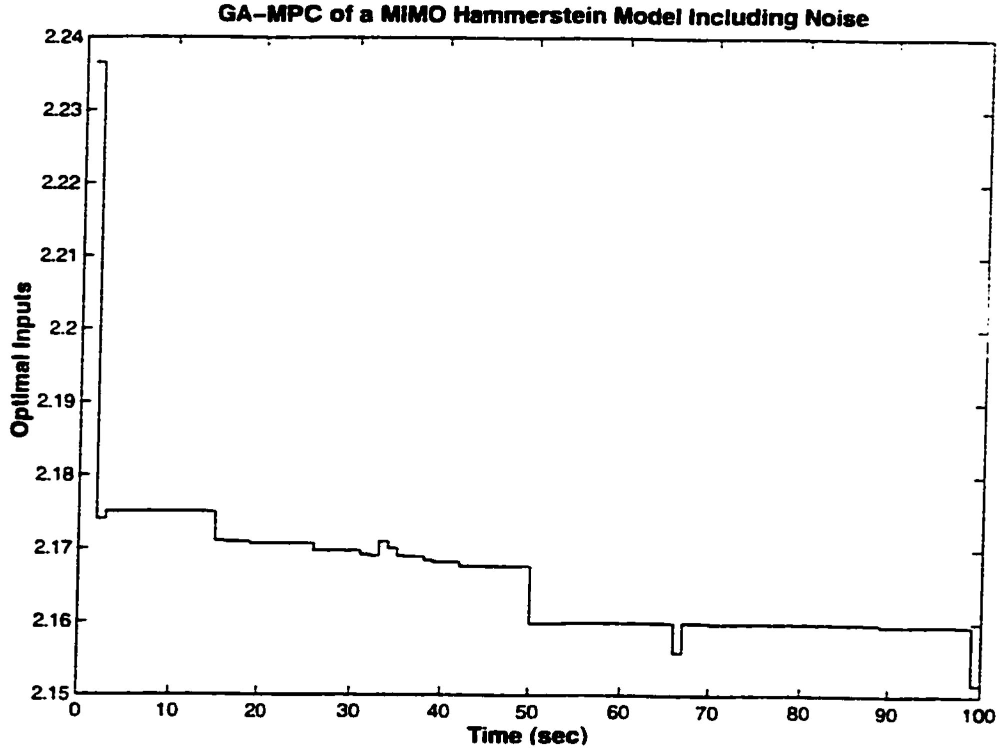

Figure 4.47: Optimal Inputs generated by the controller for a 2-input 2-output Hammerstein Model including noise (Second Input).

4.3 Adaptive GA-Based Predictive Control

It is highly unrealistic to think that the model which is to be used in a model predictive controller is equivalent to the actual process. This process model is usually made available at the controller design stage, after conducting open loop tests on the process. Performance of MPC could become unacceptable due to a very inaccurate model, thus requiring a more accurate model. Such a model frequently has to be developed while the process is kept under MPC. Therefore the need to study the model/plant mismatch is a must. This task has an element of duality in it: learning and regulation. For learning, it ensures that inputs to the process are exciting enough to yield information about the process dynamics. With regard to regulation, the process input tries to keep the process output at a desired set point. The difficulty of closed loop identification or learning is that the input of the process to be identified is not directly selected by the designer but ultimately by the controller. Therefore special control signals are needed which are used for both identification and control. There are different methods employed to encounter this problem, see for example [41], [42], and [43]. In general, there is something called *Persistant Excitation* which is a condition the inputs must meet for identification. Parameter identification is feasible if the input to the process are persistantly exciting so as to excite all the modes of the system. For this purpose external *dithering* signals are added to dither the input before applying it to the process. This is realizable in practice, because there is always present some noise on the input and output of the process.

In the next subsection, simulation results are provided for model/plant mismatch condition and the control strategy for the proposed solution is described along with simulation results.

4.3.1 Simulations

First, the performance of the proposed GA-MPC algorithm is shown for a model/plant mismatch condition. Next, the proposed Adaptive GA-MPC Controller is presented along with parameter convergence and input/output plots. The linear model from Section 4.2.1 is simulated again but now the model uncertainty is taken into account.

GA-MPC with Model/Plant Mismatch

Let the real behavior of a linear process be described by the transfer function

$$G(q^{-1}) = \frac{0.0712q^{-1} + 0.0639q^{-2}}{1 + 1.591q^{-1} - 0.726q^{-2}} \tag{4.29}$$

The process input u must satisfy the constraints $0 \leq u \leq 5$ at all times k. Assume that the linear model

$$G(q^{-1}) = \frac{0.077q^{-1} + 0.066q^{-2}}{1 + 1.55q^{-1} - 0.77q^{-2}} \tag{4.30}$$

is available for the above process from previous data. The GA-MPC algorithm is applied to the system and the results are shown in Figure 4.48. It is clear that the controller is not able to track the set point. There is always some offset present between process output and the set point which is vanished when the model matches the process exactly.

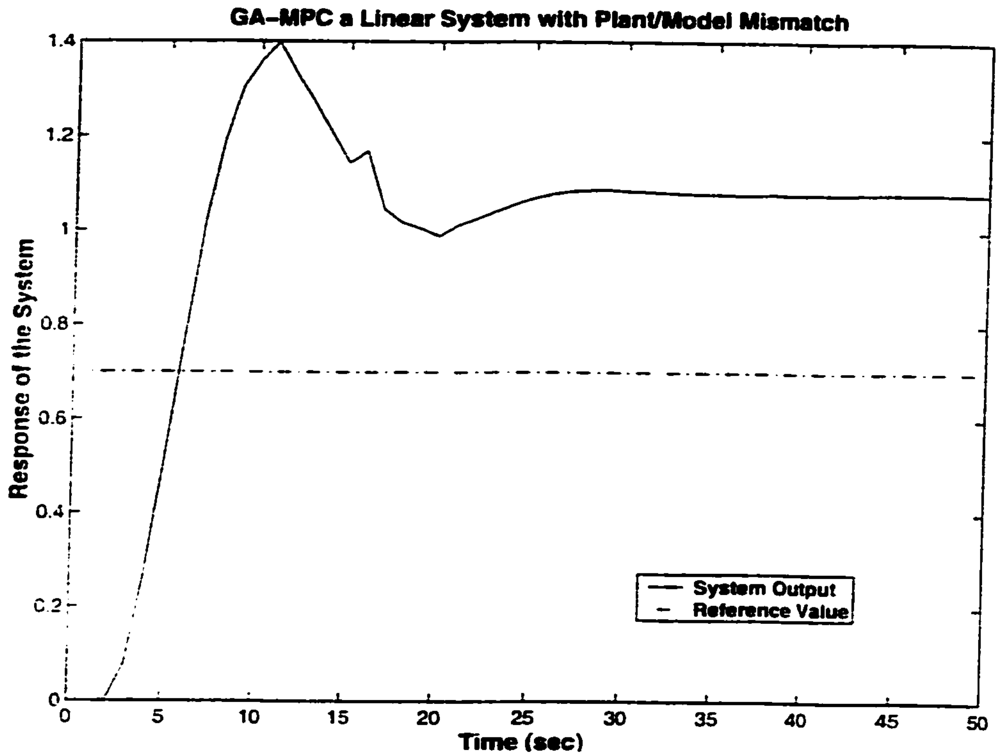

Figure 4.48: GA-MPC of Linear Process with Model/Plant Mismatch.

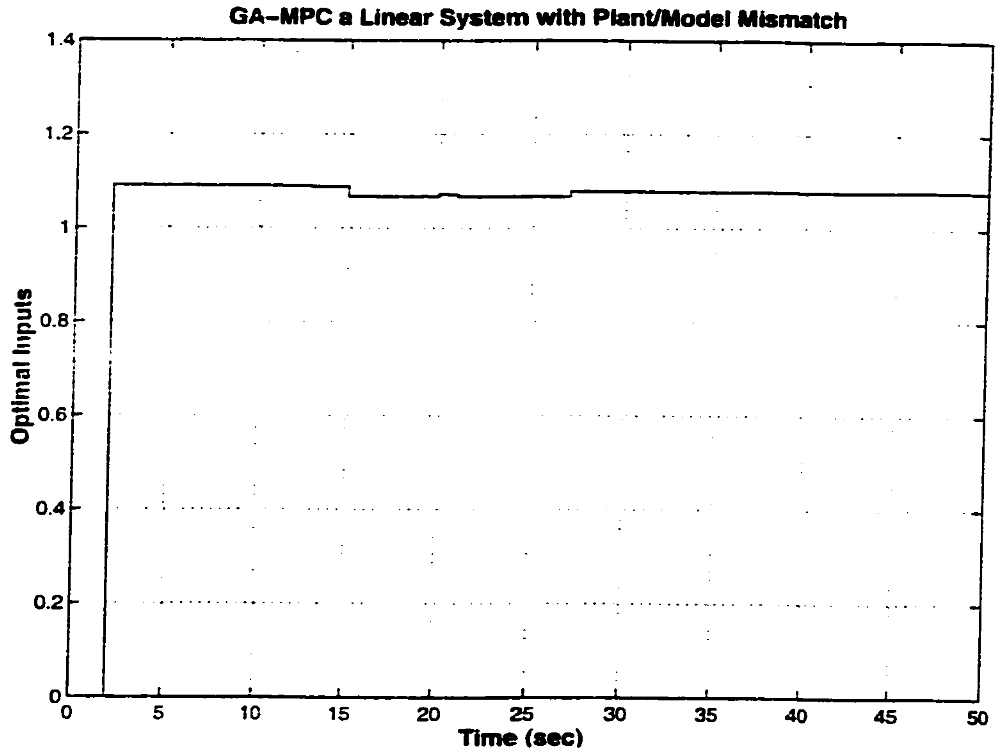

Figure 4.49: GA-MPC of a Linear Process with Model/Plant Mismatch.

Adaptive GA-MPC of a Linear Process

This section introduces the adaptive GA-MPC of a linear process. The control strategy is shown in Figure 4.50. In addition to the proposed GA-Based Model

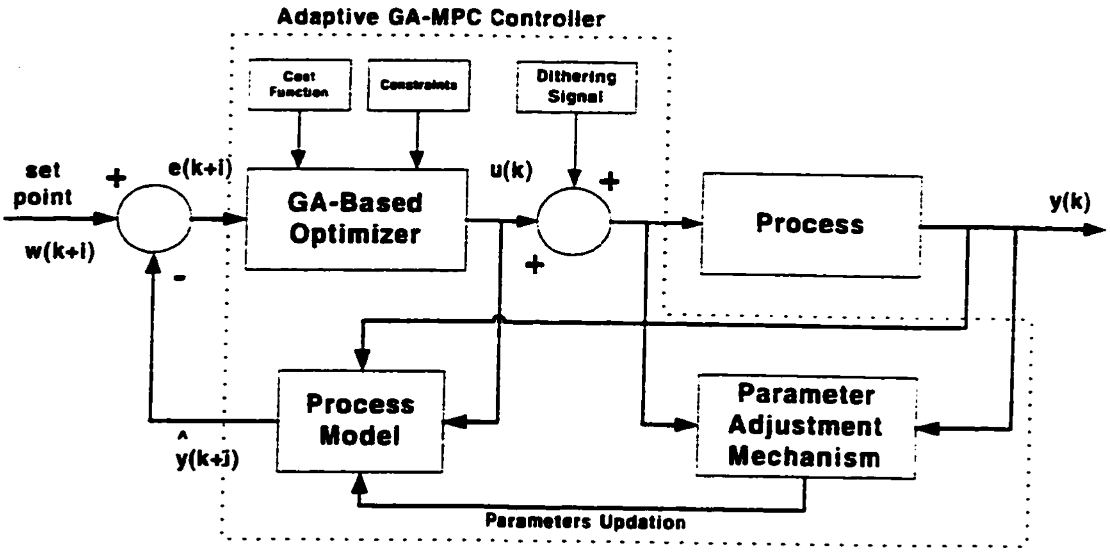

Figure 4.50: Proposed Control Strategy for Adaptive GA-MPC.

Predictive controller, there are two additional blocks. The first one is the identifier which updates the parameters of the process model. The other block is producing the *dithering signal* which is required to develop *persistant excitation* in the input signal to the process.

Consider again the real behavior of the linear process described by Equation 4.29, while its available model from previous data is given by Equation 4.30. The process input was dithered through addition of a normally distributed signal with zero mean and variance of 0.01. The parameters used in the simulation are provided in Table 4.9. Constraints imposed on the inputs are

	Population size	Number of Generations	Mutation Probability	Crossover Probability	Prediction Horizon(H_p)	Control Horizon(H_c)
Tracking	100	30	0.005	0.7	5	5

Table 4.9: Parameters used in the simulation of an Adaptive GA-MPC of a Linear System.

$$-1 \leq u \leq 1$$

The algorithm was simulated and the results are shown in Figures 4.51, 4.52 and 4.53. Figure 4.53 shows the parameters of the model converging towards their real values. Figure 4.51 is the output of the process which is oscillating around the desired set point. The oscillations are due to the dithering of input signal shown in Figure 4.52. The parameters are updated using the *Least Mean Squares* (LMS) algorithm with a learning rate of 1.2. The update equation of LMS algorithm is given by

$$W_{n+1} = W_n + \mu EX \tag{4.31}$$

where W_n is the current estimate of the parameters, W_{n+1} is the updated parameter estimate, μ is the learning rate, E is the error between the output of the process and the output of the estimated model, and X contains the regressions. For an ARMA model,

$$y(t) = \sum_{i=1}^{n} a_i y(t-i) + \sum_{j=1}^{m} b_j u(t-j) \tag{4.32}$$

or in terms of q^{-1} operator

$$y(t) = \frac{B(q^{-1})}{A(q^{-1})} q^{-d} u(t) \tag{4.33}$$

W_n, E and X are given respectively as

$$W_n = [a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_m] \tag{4.34}$$

$$E = y - \hat{y} \tag{4.35}$$

$$X = [\hat{y}(k-1), \hat{y}(k-2), \cdots, \hat{y}(k-n), u(k-1), u(k-2), \cdots, u(k-m)] \tag{4.36}$$

where a_n, b_m are the parameters to be identified. y is the output of the process. $\hat{y}$ is the predicted output of the estimated process model and u is the input to the process.

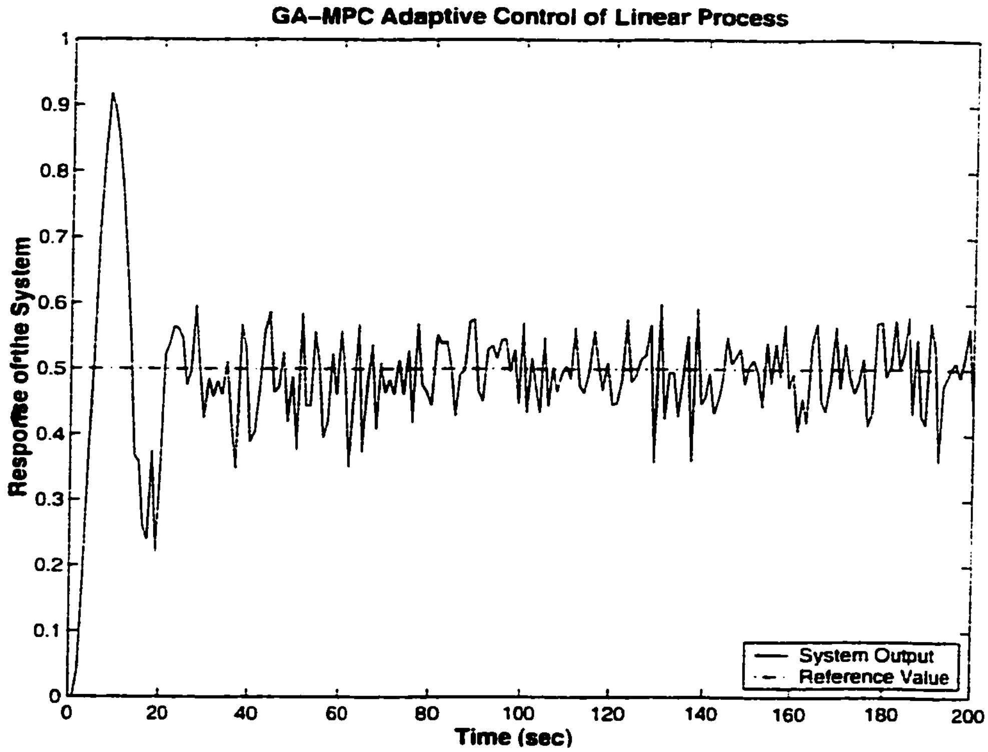

Figure 4.51: Adaptive GA-based MPC of Linear Process with Model/Plant Mismatch (Process Output).

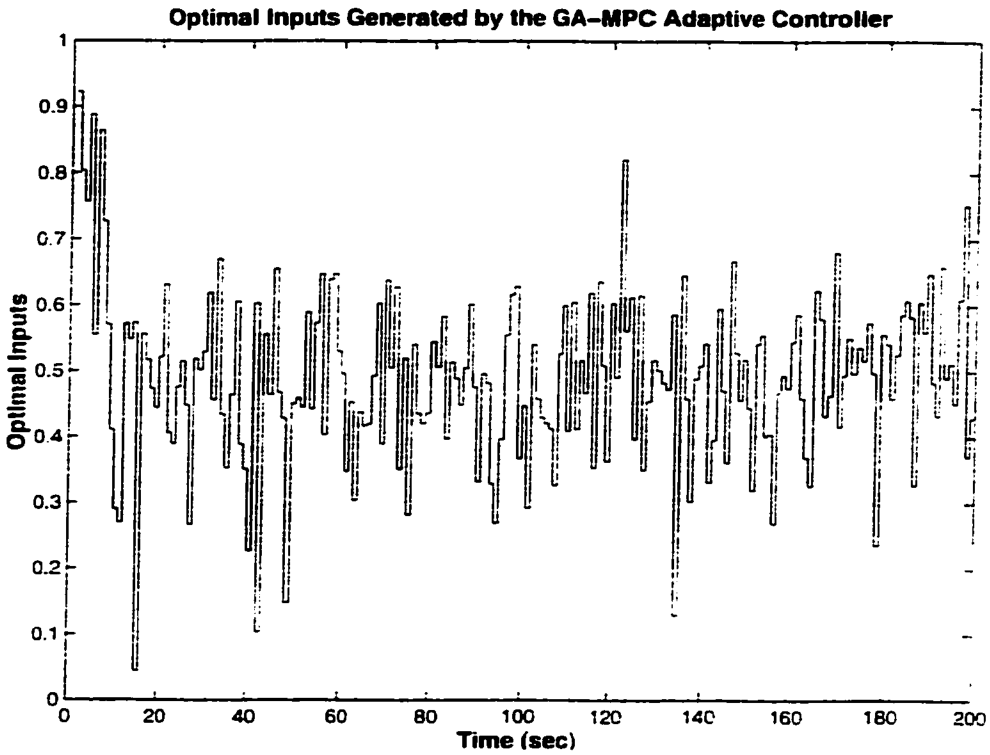

Figure 4.52: Adaptive GA-Based MPC of a Linear Process with Model/Plant Mismatch (Process Input with Dithering).

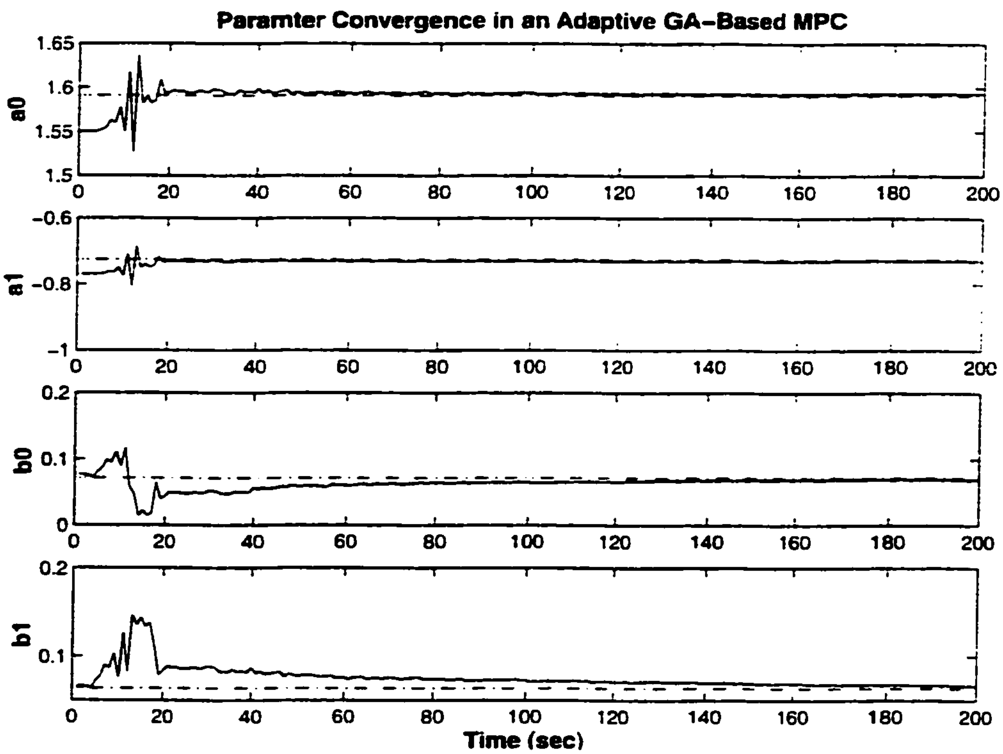

Figure 4.53: Parameter Convergence in an Adaptive GA-based MPC of a Linear Process with Model/Plant Mismatch (- · - true parameter,— estimated parameter).

4.4 Discussion

4.4.1 Effect of Prediction and Control Horizons

The prediction and control horizons play a major role in the tuning of a predictive controller. It is typically tuned by means of these two parameters. To recall, prediction horizon H_p represents the time in the future upto which predictions for the output are made, and control horizon H_c is the time in the future after which the input is held constant. In general $H_c \leq H_p$. Increasing the prediction horizon H_p has a generally stabilizing effect on the closed loop system. However, increasing H_p increases the computation load in solving the equations which must be satisfied in a least-squares sense by the controller optimization. Increasing the control horizon H_c gives more degrees of freedom to the control optimization and it may result in better control system performance. However this causes larger changes in the control variable and a reduction of the closed-loop robustness [20]. The effect of changing the control horizon on the robustness of the controller can be seen in Figures 4.54 and 4.55. The input to the controller becomes more and more sluggish as the control horizon is increased, however, the controller is still able to track the set point.

Therefore the choice of a short control horizon ($H_c =$ few units) and a large difference between H_c and H_p generally improves the stability properties of the closed-loop system [15]. Therefore, for robustness reasons it is better to use a large value for H_p. In [44], it was argued that the prediction horizon should be chosen upto the time where the output equals atleast 90 to 95% of the steady state value. A comparison table is

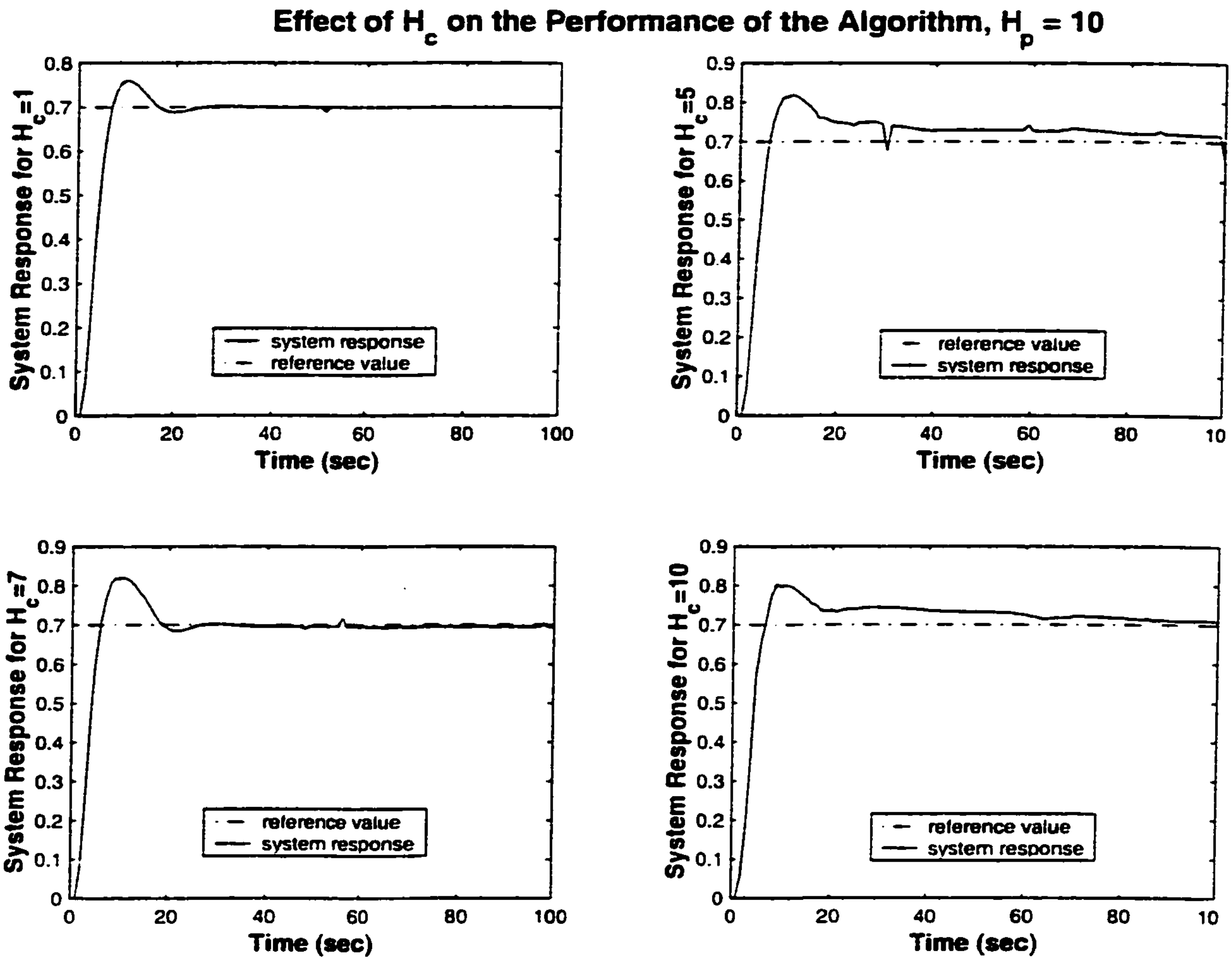

Figure 4.54: Effect of Increasing the Control Horizon on the Performance of the Control Algorithm (System Response).

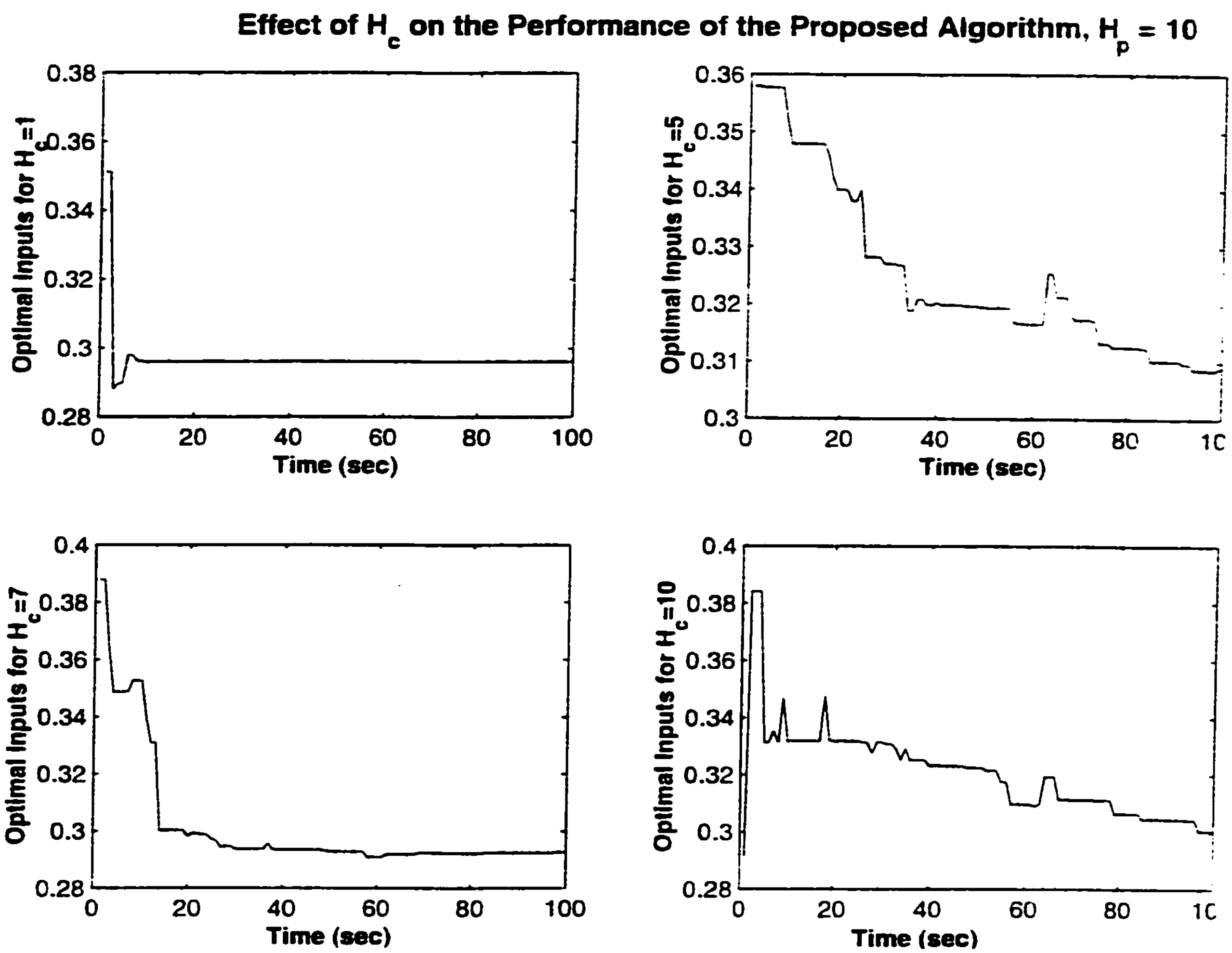

Figure 4.55: Effect of Increasing the Control Horizon on the Performance of the Control Algorithm (Optimal Inputs generated by the controller).

provided below to compare different values of prediction horizon with the standard deviation of the output error. For this purpose, white gaussian noise with zero mean and variance 0.001 is added to the output of the control valve from Section 4.2.2. The mean square error and variance of the output error is calculated for each value of prediction horizon and is shown in Table 4.10. Initially both the mean square

No.	Prediction Horizon	Mean Square Error	Variance
1	2	0.0558	0.0349
2	3	0.0389	0.0374
3	5	0.0248	0.0302
4	7	0.0245	0.0280
5	10	0.0290	0.0292
6	15	0.0281	0.0273
7	20	0.0291	0.0227

Table 4.10: Effect of the Prediction Horizon on the Output of the Process.

error and variance of the output error decreases, however, as H_p is increased beyond a certain value, the mean square error and variance becomes almost constant.

In this thesis, however, the prediction horizon and control horizon are chosen for which best results were obtained. During the course of simulations it was found that with $H_c = 1$, best results can be obtained, so in most of the simulations the control horizon is taken as 1. The value of the prediction horizon varies. From extensive simulation work it has been observed that the minimum value of the prediction horizon must be equal to the maximum delay in the system.

Thus,

$$H_{p_{min}} = \text{maximum delay in the process}$$

A simulation example is now presented for which the prediction horizon is taken less than the maximum delay in the system. The control valve from Section 4.2.2 is taken. From the linear dynamics of the control valve in Equation 4.5, it can be readily seen that the maximum delay of the system is 2. Thus a prediction horizon of 1 has been selected and the system is simulated keeping the other parameters at the same value. The input constraints are also changed to

$$0 \leq u \leq 4$$

The results are shown in Figures 4.56 and 4.57 for the output and input respectively. It is clear that the controller could not be able to track the set point.

4.4.2 Effect of GA Parameters

For the simple GA which is used throughout in this thesis, there are basically four GA parameters which can be tuned to obtain the desired performance. These are *population size*, *number of generations*, *mutation probability* and *crossover probability*. Generally these parameters are tuned by hit and trial method by performing extensive simulations. In this section, the effect of these parameters to the proposed algorithm, specifically the population size and the number of generations will be presented. Results are compared on the basis of *maximum* and *average fitness* of the population. The results presented in Tables 4.11 and 4.12 were obtained by simulating the heat exchanger from Section 4.2.2 for different values of population size and number of generations.

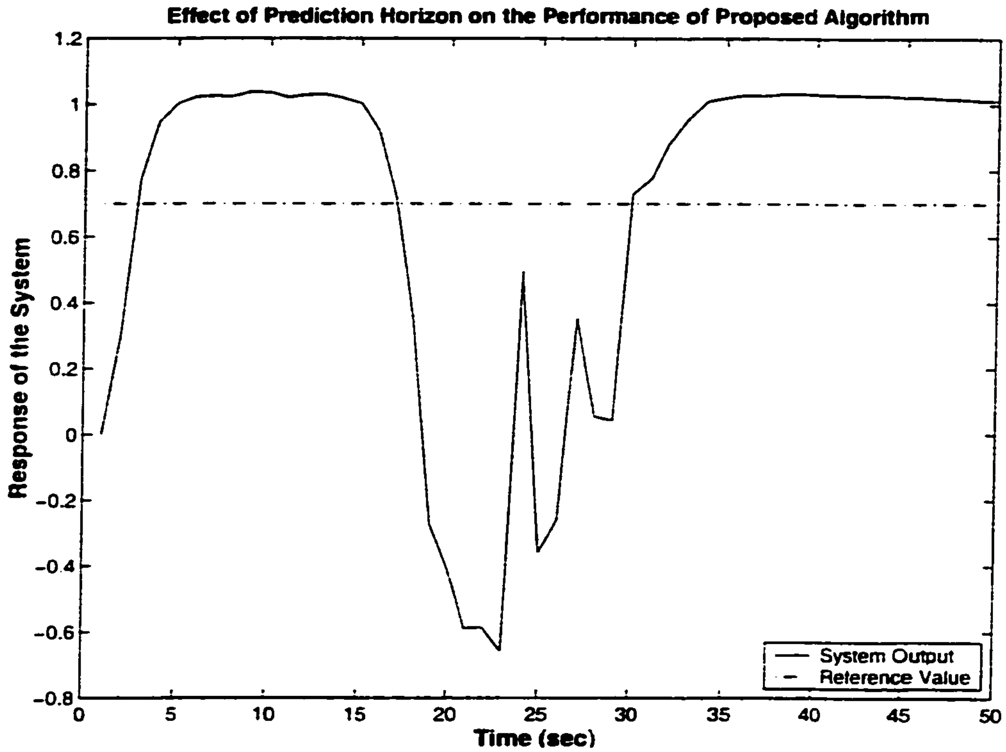

Figure 4.56: Effect of Prediction Horizon on the Performance of the GA-MPC Controller (Output Response).

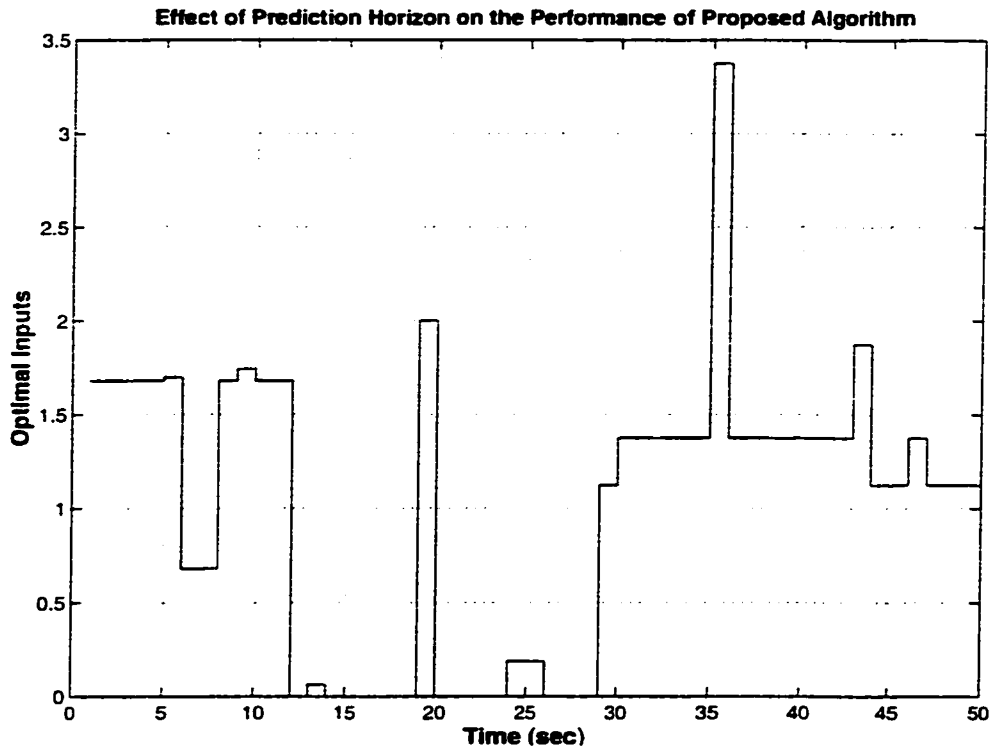

Figure 4.57: Effect of Prediction Horizon on the Performance of the GA-MPC Controller (Inputs Generated).

No.	Population Size	No. of Generations	Maxmimum Fitness	Average Fitness
1	25	100	0.0904	0.0903
2	50	100	0.0927	0.0903
3	75	100	0.1691	0.1603
4	100	100	0.5379	0.5135
5	125	100	0.7183	0.6769
6	150	100	0.8533	0.8278
7	175	100	0.8617	0.8177
8	200	100	0.9330	0.7440
9	225	100	0.9172	0.8778
10	250	100	0.9580	0.9250

Table 4.11: Effect of Population size on the performance of the Controller.

No.	Population Size	No. of Generations	Maxmimum Fitness	Average Fitness
1	75	25	0.013	0.0069
2	75	50	0.0635	0.0398
3	75	75	0.0831	0.0776
4	75	100	0.2182	0.2048
5	75	125	0.4893	0.4764
6	75	150	0.5652	0.4710
7	75	175	0.6987	0.6661
8	75	200	0.8648	0.8327
9	75	225	0.9200	0.8830
10	75	250	0.9200	0.8900

Table 4.12: Effect of Number of Generations on the performance of the Controller.

The tables shows the maximum and average fitness for different values of population size and number of generations. Both tables shows improvement in the fitness value as the population size or number of generations increases. An increase in any of the values increase the performance and the computation time. Therefore compromise between these values must be made for optimum performance.

4.4.3 Computational Complexity

There are myriad factors on which the time complexity of the proposed algorithm depends. The major is the control loop itself. Since in predictive control one has to predict future outputs of the process over a finite horizon so it is clear that the longer the prediction horizon the more is the time consumption. The computation time also depends on the GA parameters which includes population size. number of genes and number of generations. The population size represents the number of solutions available so the larger the population size the more will be the accuracy and the more is the computation time. The number of genes in this case is equivalent to the prediction horizon i.e.. the prediction horizon is represented by the number of genes in a chromosome or a solution. The number of generations or the generational loop also affects the computation time. Usually some stopping criteria is imposed on this loop, for example one can let this loop stop on the condition of achieving a pre-specified value of number of generation or it can also be discontinued when the inputs are converged or in other words. there is no further change in the solution.

Chapter 5

Summary, Conclusion and Future Work

In this thesis. Genetic Algorithms (GAs) are applied to the Model Based Predictive Controllers as an optimization tool. Nonlinear processes are mainly discussed with emphasis on chemical plants. The algorithm was applied to SISO as well as MIMO systems. A brief summary of the work done, followed by conclusions and some future extensions of this work is provided in the next section.

5.1 Summary

The main points in the thesis are

- The proposed method formulates MPC as an optimization problem and Genetic Algorithms are used in the optimization process. The proposed algorithm is called GA-MPC.

- Genetic Algorithms (GAs) are applied to nonlinear chemical processes. Different process models are used throughout the thesis. Nonlinear as well as linear processes are used. Some case studies are also dealt. An example involving a disturbance model is also considered.

- The thesis deals only with the model predictive control of processes with no eye towards the identification of the model being considered.

- Process constraints are taken into account during the optimization. Input constraints and rate of change of input constraints are considered and simulated on different processes.

- Model uncertainty is also considered and a solution is proposed for it. The resulting solution uses the Least Mean Square (LMS) algorithm to identify the model parameters. However the structure of the model is assumed to be known. The proposed solution is called Adaptive GA-Based Predictive Control.

- Single-Input Single-Output as well as Multi-Input Multi-Output processes are simulated. Case studies for both cases are considered.

- The effect of different tuning parameters is also considered. These include the basic MPC parameters i.e.. prediction and control horizons and the weighting matrices in the objective function. Effect of GA parameters is also considered and a comparison table is provided involving different values of population size and number of generations.

5.2 Conclusion

- Most of the work done on nonlinear model predictive controllers uses a linearized model to represent a nonlinear process around a certain operating point. This possesses drawback in cases, where rapid fluctuations in set point are present. The proposed algorithm deals with this problem by considering the nonlinear model, which can drive the output to any set point. The only limitation is from the input side where it is constrained due to physical limitations of actuators.

- The predictive control problem is well formulated and Genetic Algorithms (GAs) are effectively implemented to minimize the cost function subject to constraints. Real coded GAs are used throughout to get the real life flavor. The outputs seem to converge well to the desired set point.

- Real time implementation (model uncertainty) of the proposed algorithm is also considered, for which quite impressive results are obtained. However, it is assumed that the model structure is exactly known.

- Input and rate constraints are shown to be effectively imposed in the controller design. Output constraints are not considered.

- Only stable systems are considered. The algorithm was not applied to unstable systems. as the basic MPC algorithm itself is not well suited for unstable systems.

5.3 Future Work

Scientific research is an ongoing process and there is always some room for improvement. The following is a brief list of suggestions for possible future work in this area.

- In this thesis. Genetic Algorithms are used as an optimization tool to control nonlinear processes using model predictive control. Other evolutionary algorithms like Simulated Annealing. Tabu Search. Simulated Evolution etc. can also be used for this purpose.

- The basic GA operators, mutation, crossover etc. are used in their simplest form. Modified form of these operators can also be used to ensure global convergence. For example. one can use a mutation operator which is dependent on the number of generations. As the number of generations increase or in other words, the steady state value is attained, the effect of the mutation becomes negligible. This property causes this type of operator to make a uniform search in the initial space, and very locally at a later stage, favoring local tuning. An excellent review of different forms of crossover and mutation operators for real coded Genetic Algorithms can be found in for example [45].

- Only chemical processes identified by Hammerstein and Wiener Models are mainly considered in this thesis. However, the algorithm is general to any type of nonlinear constrained and unconstrained process. It can easily be modified to accomodate any type of nonlinear process.

Bibliography

[1] Pradeep B. Deshpande, editor. *"Multivariable Process Control"*. Instrument Society of America. 1989.

[2] David M. Prett Carlos E. Garcia and Manfred Morari. "Model Predictive Control: Theory and Practice — a Survey". *Automatica*, 25(3):335–348. 1989.

[3] J. L. Testud J. Richalet. A. Rault and J. Papon. "Model Predictive Heuristic Control: Applications to Industial Processes". *Automatica*. 14:413–428. 1978.

[4] C. Cutler and B. Ramaker. "Dynamic Matrix Control – a computer control algorithm". *Proceedings of the Joint Automatic Control Conference*. 1980.

[5] Mohtadi C. Clarke. D. W. and Tuff P. S. "Generalized Predictive Control. Part 1: The basic algorithm". *Automatica*. 23(2):137–148. 1987.

[6] Mohtadi C. Clarke D. W. and Tuff P. S. "Generalized Predictive Control. Part 2: Extensions and interpretations". *Automatica*, 23(2):149–160. 1987.

[7] S. Joe Qin and Thomas A. Badgewell. "An overview of nonlinear model predictive control applications". *Technical Report*.

[8] David Clarke, editor. *"Advances in Model-Based Predictive Control"*. Oxford Science Publications. 1994.

[9] J. Richalet. "Industrial Applications of Model Based Predictive Control". *Automatica*, 29(5):1251–1274, 1993.

[10] P. D. Roberts. "A Brief Overview of Model Predictive Control". *IEE*, 1999.

[11] S. Joe Qin and Thomas Badgwell. "An Overview of Industrial Model Predictive Control Technology". 1997. http://www.che.utexas.edu/qin/ps/cpcv16.ps.

[12] Madhukar Pandit Stefan Bergold and Bernd-Markus Pfeiffer. "Unified Representation of Predictive Control Algorithms for Practical Appications". In *UKACC International Conference on Control*, number 455, pages 787–792. IEE, September 1998.

[13] Jay H. Lee Manfred Morari. "Model Predictive Control: past, present and future". *Computers and Chemical Engineering*, 23:667–682, 1999.

[14] James B. Rawlings. "Tutorial Overview of Model Predictive Control". *IEEE Control Systems Magazine*, pages 38–52, June 2000.

[15] Ronald Soeterboek. *"Predictive Control, A Unified Approach"*. Prentice Hall, 1992.

[16] Edward Katende and Arthur Jutan. "Nonlinear Predictive Control". *Proceedings of the American Control Conference*, 6(6):4199–4203, 1995.

[17] A. C. Brooms M. Cannon, B. Kouvaritakis and Y. I. Lee. "Efficient Nonlinear Model Predictive Control". *Proceedings of the Americal Control Conference*, pages 255–259, June 2000.

[18] Ahmet Palazoglu Sandra J. Norquay and José A. Romagnoli. "Application of Wiener Model Predictive Control (WMPC) to an Industrial C2-Splitter". *Journal of Process Control*, 9:461–473, 1999.

[19] Ahmet Palazoglu Sandra J. Norquay and José A. Romagnoli. "Application of Wiener Model Predictive Control (WMPC) to a pH Neutralization Experiment". *IEEE Transactions on Control Systems Technology*, 7(4):437–445, July 1999.

[20] Ahmet Palazoglu Sandra J. Norquay and José A. Romagnoli. "Model Predictive Control based on Wiener Models". *Chemical Engineering Science*, 53(1):75–84, 1998.

[21] A. Palazoğlu K. P. Fruzzetti and K. A. McDonald. "Nonlinear model predictive control using Hammerstein models". *Journal of Process Control*, 7(1):31–41, 1997.

[22] Hideyuki Fujiwara et al. Yutaka Iino, Koji Tomida. "A new input/output constrained Model Predictive Control with frequency domain tuning technique and its application to an ethylene plant". *IEEE*, pages 457–462, 1993.

[23] J. B. Gomm D. L. Yu and D. Williams. "Model Predictive Control of a Chemical Process Using Neural Networks". *UKACC International Conference on Control, IEE*, pages 798–803, 1-4 Septmeber 1998.

[24] Sebastian Engell Andreas Draeger and Horst Ranke. "Model Predictive Control Using Neural Networks". *IEEE*, pages 61–66, October 1995.

[25] J. Gomez-Ortega D. R. Ramirez, D. Limon and E. F. Camacho. "Nonlinear MBPC for Mobile Robot Navigation using Genetic Algorithms". *Proceedings of IEEE, International Conference on Robotics and Automation*, pages 2452–2457, May 1999.

[26] Robert M. Dimeo and Kwang Y. Lee. "Genetic-Based Control of a MIMO Boiler-Turbine Plant". *IEEE Proceedings of the 33rd Conference on Decision and Control.* pages 3512–3517. December 1994.

[27] Yun Li and Kim Chwee Ng. "Design of SISO sliding mode control systems using Genetic Algorithms".

[28] A. Chipperfield and P. Fleming. "Gas-Turbine Engine Controller Design using Multiobjective Genetic Algorithms". *IEE Genetic Algorithms in Engineering Systems: Innovations and Applications.* pages 214–219. September 1995.

[29] Sadiq M. Sait and Habib Youssef. *"Iterative Computer Algorithms with Applications in Engineering. Solving Combinatorial Optimization Problems"*. IEEE Computer Society. 1999.

[30] Simone Loureiro de Oliveira. *"Model Predictive Control for constrained nonlinear systems"*. PhD thesis. California Institute of Technology (Caltech). California. USA. 1996.

[31] Xu Cheng and Bruce H. Krogh. "Stability constrained Model Predictive Control for nonlinear systems". *Proceedings of the 36th Conference on Decision and Control. San Diego. California USA.* pages 2091–2096. December 1997.

[32] J. A. Rossiter and B. Kouvaritakis. "Constrained Stable Generalised Predictive Control". *IEE Proceedings.* pages 243–254. July 1993.

[33] Ramine Rouhani and Raman K. Mehra. "Model Algorithmic Control (MAC): Basic Theoretical Properties". *International Federation of Automatic Control.* pages 401–414. 1982.

[34] M. Morari P. Lundstrom. J. H. Lee and S. Skogestad. "Limitation of Dynamic Matrix Control". *Computers chem. Engng..* 19(4):409–421. 1995.

[35] R. A. Hess and Y. C. Jung. "Generalized Predictive Control of Dynamic Systems". *IEEE.* pages 844–849. 1988.

[36] Chi-Tsong Chen. *"Linear System, Theory and Design 3ed"*. Oxford University Press, 1999.

[37] Stanley H. Johnson Esref Eskinat and William L. Luyben. "Use of Hammerstein Models in Identification of Nonlinear Systems". *AIChE.* 37(2):255–268, 1991.

[38] Torbjorn Wigren. *"Recursive Identification Based on the Nonlinear Wiener Model"*. PhD thesis, ACTA Universitatis Upsaliensis. 1990.

[39] R. Zouhal and N. E. Radhy. "Adaptive Control of Multivariable Hammerstein's Nonlinear Model". *IEEE.* pages 147–150.

[40] H.F.Wang. "Philips-Heffron Model of Power Systems Installed with STATCOM and applications". *IEE Proceedings on Generation. Transmission and Distribution*, pages 521–527. September 1999.

[41] Premkiran Vuthandam Manoj Shouche. Hasmet Genceli and Michael Nikolaou. "Simultaneous constrained model predictive control and identification of DARX processes". *Automatica*. 1998.

[42] S. Alper Eker Alexander Schwarm and Michael Nikolaou. "Model Predictive Control and Identification: a new approach to closed loop identification and adaptive control".

[43] Michael Nikolaou and S. Alper Eker. "Adaptive control through online optimization: The MPCI paradigm and variants". *AMCS*. 1998.

[44] *"Advanced Control Systems"*. 1994.

[45] M. Lozano F. Herrera and J. L. Verdegay. "Tackling Real Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis". *Department of Computer Science and Artficial Intelligence, University of Granada, Spain.*

Vitae

- Wasif Naeem.
- Born in Karachi, Pakistan.
- Received Bachelor of Engineering (B.E.) degree in Electrical Engineering from NED University of Engineering and Technology, Karachi, Pakistan in July 1998.
- Worked as a Research and Development Engineer in Laser Manufacturing in Karachi, Pakistan from October 1998 to July 1999.
- Joined the Department of Electrical Engineering at KFUPM as a Research Assistant in September 1999.
- Completed Master of Science (M.S.) in Electrical Engineering at KFUPM in May 2001.
- Email: w_naeem@yahoo.com